Visual Imitation Learning of Manipulation Tasks for Humanoid Robots

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte
DISSERTATION

von

M.Sc. Jianfeng Gao

aus Shandong China

Tag der mündlichen Prüfung:	08.05.2025
Erster Gutachter:	Prof. Dr.-Ing. Tamim Asfour
Zweiter Gutachter:	Prof. Dr. Marc Toussaint

Abstract

Observational learning is a fundamental mechanism by which humans acquire new skills by watching others and understanding the consequences of their actions. This capability allows for skill acquisition through demonstration, thereby reducing the need for costly trial-and-error processes. Cognitive development research has shown that infants can learn complex skills and make inductive generalizations from sparse samples by observing caregivers and peers; they leverage statistical evidence that models the covariation of task features, all without direct physical interaction or explicit linguistic instructions. By identifying invariant task features – such as keypoints associated with an object's functional parts – from high-dimensional visual inputs, it is possible to derive effective and transferable task representations. These insights have motivated significant research in robotics to develop *Visual Imitation Learning* (VIL) systems that emulate human observational learning mechanisms. Nevertheless, acquiring generalizable task representations solely from sparse human demonstration videos remains a significant challenge.

In this thesis, we adopt a bottom-up approach that extracts essential invariant task features from demonstrations without relying on ground-truth labels, direct physical interaction, or linguistic bootstrapping commonly employed in top-down methodologies. We address the computational challenges of learning complex manipulation tasks by decomposing them into tractable sub-problems in both spatial and temporal domains. In spatial domain, our approach extracts object-centric, keypoint-based geometric constraints that capture the functional aspects of objects and spatial coordination between arms. We then leverage neural object descriptors to facilitate the transfer of learned tasks to novel object

instances or categories. In the temporal domain, we segment human demonstrations hierarchically and learn temporal coordination and action primitives. Throughout the work, we employ variance-based statistical analyses to extract invariant task features – including keypoints, common viewpoints, object and hand dominance, and spatio-temporal constraints – from sparse human demonstrations. This research addresses the following key research questions: 1) How can keypoint-based subsymbolic task representations be modeled to enable intra-category generalization? 2) How can these representations be effectively detected and extracted from visual input? 3) How can coordination strategies for various unimanual and bimanual manipulation be learned and incorporated into bimanual compliance controllers? 4) How can demonstration videos be segmented at a consistent granularity level to facilitate learning of spatio-temporal coordination?

Learning Keypoint-based Task Representation

We first focus on modeling generalizable subsymbolic *keypoint-based task representations* and learning them from sparse human demonstration videos of unimanual manipulation tasks. To this end, we propose a neural descriptor-based object representation along with a perception pipeline for VIL that detects humans and objects, tracks their states, proposes keypoint candidates, and establishes dense correspondences between object instances to address viewpoint mismatches and object variations. Our *Principal Constraint Estimation* (PCE) algorithm extracts sparse yet semantically meaningful keypoints – associated with object functional parts – from the candidate set by analyzing the statistical variance of their spatial distribution across multiple demonstrations. PCE simultaneously derives *keypoint-based geometric constraints on principal manifolds*, their corresponding *local frames*, and *movement primitives* as subsymbolic task representations. In contrast to most existing approaches that learn only a subset of these representations, our method provides a comprehensive understanding of task constraints. The resulting task representations are interpretable, transferable, viewpoint invariant, and embodiment-independent. Consequently, the learned tasks can be robustly generalized to novel object categories and reproduced by a novel keypoint-based admittance controller on a humanoid robot. Our key insight is that a sparse, object-centric representation combined with dense correspondence detection greatly enhances intra-category generalization,

enabling the learning of various daily tasks from as few as 10 demonstration videos in cluttered scenes.

Learning Bimanual Coordination Strategies

Bimanual manipulation introduces additional complexity due to intricate object relationships, fine-grained motion details, and diverse coordination strategies between the arms. Similarly to unimanual tasks, bimanual manipulation tasks exhibit invariant features across multiple demonstrations. We extend our unimanual task representations to the bimanual domain by introducing a novel hybrid master-slave object relationship, which encapsulates the roles of both objects and hands in a task by exploiting statistical invariances in their spatial distributions. This formulation enables the derivation of various coordination strategies that cover a complete taxonomy of bimanual manipulation tasks, thereby unifying uni- and bimanual task representations. Fine-grained, keypoint-based geometric constraints enable our approach to capture detailed motion styles from demonstrations, paving the way for modeling personalized task representations in the future. Based on the extracted bimanual coordination categories, we develop real-time compliance controllers designed to manage bimanuality, motion synchronization, and hybrid master-slave relationships.

Keypoint-based Segmentation, Bimanual Coordination and Grasping

Human demonstrations typically consist of a sequence of actions, making the detection of common motion segments across demonstrations crucial for learning comprehensive task representations. To address this, we propose a *keypoint-based hierarchical motion segmentation algorithm* for VIL that leverages the motion characteristics of keypoints within object-centric local frames. By exploiting dense keypoint information, our algorithm accurately identifies changes in the static and dynamic spatial relationships of objects at the finest granularity. These fine-grained segments are then merged into semantically meaningful action primitives using derived contextual information. This bottom-up approach yields motion segments at consistent granularity levels across different layers, facilitating precise semantic and temporal alignment of segments across multiple demonstrations and enabling the learning of *spatio-temporal bimanual coordination*

strategies. As an application, we present a *task-oriented grasp learning and generation* framework that models task-specific grasp poses in the grasping segments of human demonstrations as a pose constraint relative to object functional parts, and transfers it to novel categorical objects during inference time.

Together, these three interconnected components constitute our bottom-up *keypoint-based visual imitation learning* (KVIL) framework, which derives subsymbolic spatio-temporal task representations from sparse human demonstration videos for both unimanual and bimanual manipulation tasks. The main objective of this thesis is to model invariant task features based on keypoints, geometric constraints, task roles of objects and hands, and spatio-temporal coordination – while also developing effective mathematical methods to extract these features from video inputs. The developed framework is evaluated across various daily tasks on humanoid robots, demonstrating its effectiveness and potential to robustly generalize learned tasks to novel objects and environments, thereby significantly advancing the state-of-the-art in visual imitation learning. Furthermore, this thesis opens numerous avenues for future research, including the exploration of more complex spatio-temporal task models, application to articulated and soft objects, and the development of more sophisticated learning algorithms to enhance the robustness and generalization of visual imitation learning systems to real-world scenarios.

Deutsche Zusammenfassung

Lernen aus Beobachtung ist ein grundlegender Mechanismus, durch den Menschen neue Fähigkeiten erwerben, indem sie andere beobachten und die Konsequenzen ihres Handelns verstehen. Diese Fähigkeit ermöglicht den Erwerb von Fertigkeiten durch Demonstration und reduziert so den Bedarf an kostspieligen Versuch-und-Irrtum-Prozessen. Forschung zur kognitiven Entwicklung haben gezeigt, dass Säuglinge komplexe Fertigkeiten erlernen und induktive Verallgemeinerungen aus spärlichen Beispielen ableiten können, indem sie Bezugspersonen beobachten. Dabei nutzen sie statistische Evidenz, die die Kovariation von Aufgabenmerkmalen modelliert, ohne dabei direkte physische Interaktion oder explizite sprachliche Instruktionen zu benötigen. Durch die Identifizierung invarianten Aufgabenmerkmale – wie etwa Schlüsselpunkte, die mit den funktionalen Teilen eines Objekts assoziiert sind – aus hochdimensionalen visuellen Eingaben ist es möglich, effektive und übertragbare Aufgabenrepräsentationen abzuleiten. Diese Erkenntnisse haben zu bedeutenden Forschungsfragen in der Robotik geführt, um *Visual Imitation Learning* (VIL) Systeme zu entwickeln, die Mechanismen des Lernen aus Beobachtung bei Menschen nachahmen. Dennoch stellt der Erwerb generalisierbarer Aufgabenrepräsentationen allein aus spärlichen menschlichen Demonstrationsvideos eine erhebliche Herausforderung dar.

In dieser Dissertation verfolgen wir einen Bottom-up-Ansatz, der wesentliche invariante Aufgabenmerkmale aus Demonstrationen extrahiert, ohne auf Ground-Truth-Labels, direkte physische Interaktion oder sprachliches Bootstrapping, wie sie in Top-down-Methoden häufig genutzt werden, angewiesen zu sein.

Wir adressieren die Herausforderungen beim Erlernen komplexer Manipulationsaufgaben, indem wir diese in handhabbare Teilprobleme in den räumlichen und zeitlichen Domänen zerlegen. Im räumlichen Bereich extrahiert unser Ansatz objektzentrierte, schlüsselpunktbasierte geometrische Einschränkungen, die die funktionalen Aspekte von Objekten sowie die räumliche Koordination zwischen den Händen erfassen. Anschließend nutzen wir neuronale Objektdeskriptoren, um den Transfer erlernter Aufgaben auf neuartige Objektinstanzen oder Objektkategorien zu erleichtern. Im zeitlichen Bereich segmentieren wir menschliche Demonstrationen hierarchisch und lernen zeitliche Koordination sowie Aktionsprimitive.

Während der gesamten Arbeit wenden wir varianzbasierte statistische Analysen an, um invariante Aufgabenmerkmale – einschließlich Schlüsselpunkten, gemeinsamen Blickwinkeln, der Dominanz von Objekten und Händen sowie raum-zeitlichen Einschränkungen – aus spärlichen menschlichen Demonstrationen zu extrahieren. Diese Forschung adressiert die folgenden zentralen Forschungsfragen: 1) Wie können schlüsselpunktbasierte subsymbolische Aufgabenrepräsentationen modelliert werden, um eine Intra-Kategorie-Generalisation zu ermöglichen? 2) Wie können diese Repräsentationen effektiv aus visuellen Eingaben erkannt und extrahiert werden? 3) Wie können Koordinationsstrategien für verschiedene einhändige und beidhändige Manipulationen erlernt und in bimanuale Compliance-Controller integriert werden? 4) Wie können Demonstrationsvideos auf einer konsistenten Granularitätsebene segmentiert werden, um das Erlernen räumlicher und zeitlicher Koordination zu erleichtern?

Erlernen schlüsselpunktbasierter Aufgabenrepräsentationen

Zunächst konzentrieren wir uns auf die Modellierung generalisierbarer subsymbolischer *schlüsselpunktbasierter Aufgabenrepräsentationen* und deren Erlernen aus spärlichen menschlichen Demonstrationsvideos von einhändigen Manipulationsaufgaben. Hierzu schlagen wir eine neuronale, deskriptorbasierte Objektrepräsentation in Verbindung mit einer Wahrnehmungspipeline für VIL vor, die Menschen und Objekte erkennt, deren Zustände verfolgt, Schlüsselpunktkandidaten vorschlägt und dichte Korrespondenzen zwischen Objektinstanzen herstellt, um Blickwinkeldiskrepanzen und Objektvariationen zu adressieren. Unser *Principal Constraint Estimation* (PCE)-Algorithmus extrahiert spärliche,

jedoch semantisch aussagekräftige Schlüsselpunkte – welche mit den funktionalen Teilen eines Objekts assoziiert sind – aus dem Kandidatensatz, indem er die statistische Varianz ihrer räumlichen Verteilung über mehrere Demonstrationen analysiert. PCE leitet gleichzeitig *schlüsselpunktbasierte geometrische Einschränkungen auf Hauptmannigfaltigkeiten*, deren zugehörige *lokale Bezugssysteme* sowie *Bewegungsprimitive* als subsymbolische Aufgabenrepräsentationen ab. Im Gegensatz zu den meisten bestehenden Ansätzen, die nur einen Teil dieser Repräsentationen erlernen, bietet unsere Methode ein umfassendes Verständnis der Aufgabenbeschränkungen. Die resultierenden Aufgabenrepräsentationen sind interpretierbar, übertragbar, blickwinkelinvariant und unabhängig vom Embodiment. Folglich können die erlernten Aufgaben robust auf neuartige Objektkategorien generalisiert und von einem neuartigen schlüsselpunktbasierten Admittanz-Controller auf einem humanoiden Roboter reproduziert werden. Unsere zentrale Erkenntnis ist, dass eine spärliche, objektzentrierte Repräsentation kombiniert mit dichten Korrespondenzerkennungen die Intra-Kategorie-Generalisation erheblich verbessert und das Erlernen verschiedener Alltagsaufgaben aus bereits wenigen (z.B. 10) Demonstrationsvideos in komplexen Szenen ermöglicht.

Erlernen von bimanualen Koordinationsstrategien

Beidhändige Manipulationen stellen aufgrund komplexer Objektbeziehungen, feingranularer Bewegungsdetails und diverser Koordinationsstrategien zwischen den Händen eine zusätzliche Herausforderung dar. Ähnlich wie bei einhändigen Aufgaben weisen auch beidhändige Manipulationsaufgaben invariante Merkmale über mehrere Demonstrationen hinweg auf. Wir erweitern unsere einhändigen Aufgabenrepräsentationen Aufgaben, indem wir eine neuartige hybride Master-Slave-Objektbeziehung einführen, die die Rollen von Objekten und Händen in einer Aufgabe durch die Ausnutzung statistischer Invarianzen in deren räumlichen Verteilung kapselt. Diese Formulierung ermöglicht die Ableitung diverser Koordinationsstrategien, die eine vollständige Taxonomie von beidhändigen Manipulationsaufgaben abdecken, und vereinheitlicht damit einhändige und beidhändige Aufgabenrepräsentationen. Feingranulare, schlüsselpunktbasierte geometrische Einschränkungen ermöglichen es unserem Ansatz, detaillierte Bewegungsstile aus Demonstrationen zu erfassen, was den Weg für die Modellierung personalisierter Aufgabenrepräsentationen in der Zukunft ebnet. Basierend auf den extrahierten bimanualen Koordinati-

onskategorien entwickeln wir Echtzeit-Compliance-Controller, die in der Lage sind, Beidhändigkeit, Bewegungssynchronisation und hybride Master-Slave-Beziehungen zu steuern.

Schlüsselpunktbasierte Segmentierung, bimanuale Koordination und Greifen

Menschliche Demonstrationen bestehen typischerweise aus einer Abfolge von Aktionen, wodurch die Erkennung gemeinsamer Bewegungssegmente über verschiedene Demonstrationen hinweg entscheidend für das Erlernen umfassender Aufgabenrepräsentationen ist. Zur Bewältigung dieser Herausforderung schlagen wir einen *schlüsselpunktbasierten hierarchischen Bewegungssegmentierungsalgorithmus* für VIL vor, der die Bewegungseigenschaften von Schlüsselpunkten in objektzentrierten lokalen Bezugssystemen ausnutzt. Durch die Nutzung dichter Schlüsselpunktinformationen identifiziert der Algorithmus Veränderungen in den statischen und dynamischen räumlichen Beziehungen von Objekten auf der feinsten Granularitätsebene. Diese feingranularen Segmente werden anschließend mithilfe abgeleiteter kontextueller Informationen zu semantisch bedeutungsvollen Aktionsprimitive zusammengeführt. Dieser Bottom-up-Ansatz liefert Bewegungssegmente mit konsistenter Granularität über verschiedene Ebenen hinweg, was eine präzise semantische und zeitliche Ausrichtung der Segmente über mehrere Demonstrationen hinweg ermöglicht und das Erlernen von *raum-zeitlichen bimanualen Koordinationsstrategien* unterstützt. Als Anwendung präsentieren wir ein *aufgabenorientiertes Rahmenwerk für das Lernen von Griffen*, das aufgabenbezogene Greifpositionen in den Greifsegmenten menschlicher Demonstrationen als Pose-Einschränkung in Bezug auf die funktionalen Teile von Objekten modelliert und diese während der Inferenz auf neuartige, kategoriale Objekte überträgt.

Zusammen bilden diese drei miteinander verbundenen Komponenten unseren Bottom-up-Ansatz des *schlüsselpunktbasierten visuellen Imitationslernens* (KVIL), der subsymbolische raum-zeitliche Aufgabenrepräsentationen aus spärlichen menschlichen Demonstrationsvideos für einhändige und beidhändige Manipulationsaufgaben ableitet. Das Hauptziel dieser Dissertation ist es, invariante Aufgabenmerkmale basierend auf Schlüsselpunkten, geometrischen Einschränkungen, Aufgabenrollen von Objekten und Händen sowie räumlich-zeitlicher Koordination zu modellieren und gleichzeitig effektive Methoden zur Extraktion dieser Merkmale aus Videodaten zu entwickeln. Das entwickelte Framework wird

anhand verschiedener Alltagsaufgaben auf humanoiden Robotern evaluiert, wobei seine Effektivität und sein Potenzial zur robusten Generalisierung erlernter Aufgaben auf neuartige Objekte und Umgebungen demonstriert werden. Dies stellt einen signifikanten Fortschritt im Stand der Technik des visuellen Imitationslernens dar. Darüber hinaus eröffnet diese Dissertation zahlreiche Perspektiven für zukünftige Forschungen, darunter die Erforschung komplexerer räumlich-zeitlicher Aufgabenmodelle, die Anwendung auf artikulierte und weiche Objekte sowie die Entwicklung anspruchsvollerer Lernalgorithmen zur Verbesserung der Robustheit und Generalisierbarkeit visueller Imitationslernensysteme in realen Szenarien.

Acknowledgment

First and foremost, I express my deepest gratitude to my supervisor, Prof. Dr.-Ing. Tamim Asfour, for his invaluable guidance, continuous support, and insightful feedback. I am truly grateful for the opportunity to work under his supervision. He supported me throughout this journey, granting me the freedom to explore my own ideas while ensuring I remained focused on my goals. I would also like to thank Prof. Dr. Marc Toussaint for co-supervising this thesis and providing valuable feedback that strengthened this work.

A heartfelt thank you goes to my colleagues and fellow researchers at H^2T; our discussions and collaborations significantly enriched my research. I specifically want to acknowledge Dr. Noémie Jaquier, whose high standards served as a role model for my own research skills. Additionally, I am also grateful to Zhi Tao, who has been both a great collaborator and a friend. His brainstorming sessions and contributions were invaluable during the early stages of my research on imitation learning.

Finally, I would like to thank my wife, Yan, for her unwavering support and patience. I am deeply appreciative of her willingness to discuss my research; her objective, 'third-person' perspective inspired several ideas that became a crucial part of this research. Last but certainly not least, I am deeply thankful to my parents for their constant encouragement from China and for always supporting me in pursuing what I love.

Contents

CHAPTER 1

Introduction

Social learning theory, a cornerstone in psychology and cognitive development introduced in Bandura (1977), elucidates how individuals acquire new behaviors, attitudes, and emotional responses through observation rather than direct experience or reinforcement. By integrating observation, imitation, and modeling as essential learning mechanisms, this theory bridges the gap between behaviorist and cognitive learning paradigms, offering a foundation for understanding how social interactions shape learning and behavior.

At its core, social learning theory emphasizes cognitive processes such as attention, retention, reproduction, and motivation. Extensive research in cognitive development has shown that observational learning (Burke et al., 2010) plays a crucial role in human development, beginning in infancy. Infants begin to acquire complex skills by observing and mimicking the actions of caregivers and peers. This learning process is not merely a passive mirroring of behaviors; instead, it involves active cognitive engagement, such as understanding human intentions (Meltzoff, 1995), comprehending object functional analogies (Waismeyer et al., 2015; Yiu and Gopnik, 2023; Yiu et al., 2024), predicting outcomes (Meltzoff and Prinz, 2002), and making inductive inferences (Gweon et al., 2010; Schulz, 2012; Meltzoff et al., 2012; Waismeyer et al., 2015) from visual perception alone.

Research has revealed that infants can learn inductive generalizations from remarkably sparse samples (Gweon et al., 2010; Gweon and Schulz, 2011; Schulz, 2012). Several key factors contribute to this ability. One crucial aspect is the

modeling of observed statistical evidence from demonstrated samples and the assessment of the strength of this evidence. Such evidence encompasses object properties, event probabilities and their sampling processes, invariant features, and spatio-temporal action constraints. This cognitive process enables infants, as young as nine months old, to project task properties across object instances that share semantic labels and/or visual appearances, generalize functionalities to visually similar objects (Gweon et al., 2010), and reproduce demonstrated actions on these objects. Further studies indicate that despite early-stage scale errors, infants can perform the same general action on a class of objects and calibrate their movements accordingly (DeLoache et al., 2004). This finding suggests that modeling manipulation skills in an abstract object space and conditioning actions on constraints derived from object properties are beneficial for generalization. Notably, infants efficiently acquire such inference generalization from covariation in a few demonstrations without physical interaction (trial-and-error) and without causal linguistic descriptions from demonstrators, i. e., without linguistic bootstrapping (Meltzoff et al., 2012; Waismeyer et al., 2015).

Insights from the cognitive development research on infant observational learning have significantly influenced robotics research, particularly in the field of *Visual Imitation Learning* (VIL). By emulating early human learning processes, humanoid robots can efficiently replicate not only the physical actions of humans but also their relationships to the environment, such as the connection between actions and functional parts of objects. For instance, by observing a few videos of people pouring water from kettles into teacups as shown in Figure 1.1, the task of visual imitation learning is to determine "what" a pouring task entails, "how" to perform it, and when to "when" to execute it. This task becomes particularly challenging when performed under varying conditions: using one arm (Figure 1.1a), using both arms from different viewpoints with diverse cups and object poses (Figure 1.1b), or performing a sequence of actions (Figure 1.1c).

From a computational perspective, learning complex manipulation tasks can be decomposed into simpler problems in both spatial and temporal domain. Specifically, geometric task constraints can be framed as a set of keypoint-based geometric relationships between object parts of object pairs. For instance, in a pouring task, the spout and handle of a kettle can be represented as keypoints $\mathbf{k}_1$ and $\mathbf{k}_2$, or as key regions surrounding them. These keypoints or key regions, as invariant features, facilitate aligning multiple demonstrations of the same task into a common viewpoint as shown in Figure 1.2a. This alignment allows the pouring task to be conceptualized as a set of geometric constraints as illustrated in Figure 1.2b: aligning the spout $\mathbf{k}_1$ with a point above the cup's rim (point-to-

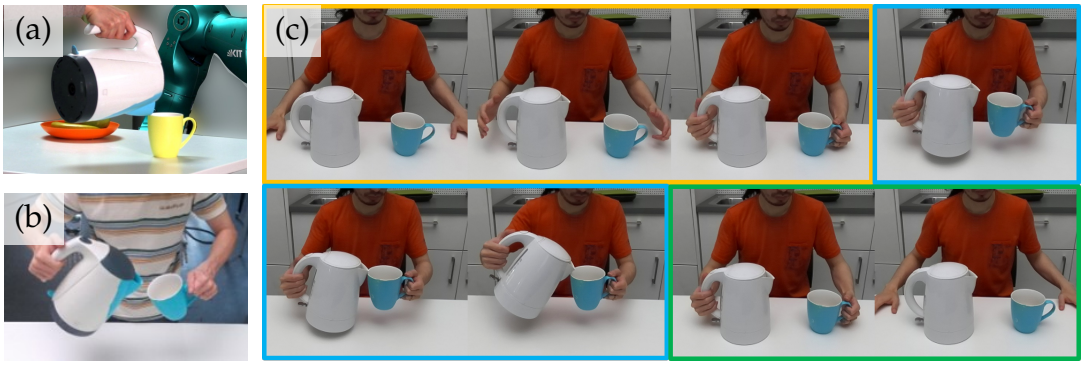
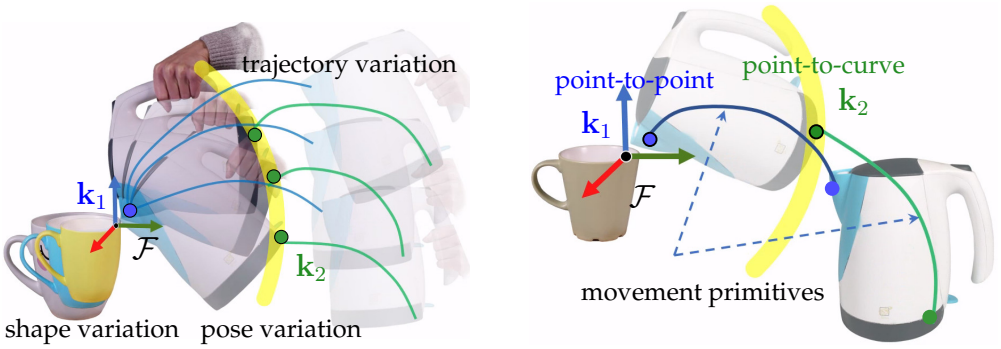


Figure 1.1.: Human demonstrations of the pouring task. (a) shows the unimanual pouring demonstration, (b) shows bimanual variants recorded from a different viewpoints, (c) illustrates the coarse sub-actions in the pouring task, including reaching ( ), pouring ( ) and placing ( ).

point constraint) and orienting the handle $\mathbf{k}_2$ to a curve that controls the kettle's inclination angle (point-to-curve constraint). This framework of keypoints, key regions, and geometric constraints can be generalized to represent a wide array of daily manipulation tasks. By parameterizing the motion of object functional parts relative to local frames of reference attached to objects, we create a flexible, object-centric representation that can adapt to various scenarios.

In the temporal domain, a manipulation tasks consisting of a sequence of sub-actions can be decomposed into simpler motion segments (Graybiel, 1998), where each segment can be learned as a primitive in an action library (Pastor et al., 2009). For instance, the pouring task comprises approach, pour and place actions at the coarse level (see Figure 1.1c). The segmentation granularity depends on task complexity and coordination requirements. When bimanual coordination strategies vary across task phases, temporal and spatial coordination must be extracted to reflect the demonstrated motion styles.

To automatically extract these keypoint-based task representations from sparse demonstrations, invariant features across multiple demonstrations must be identified: 1) Consistent visual features of object functional parts across different object instances or categories (e. g., the container openings); 2) Geometric constraints that become apparent when the same object across multiple demonstrations are spatially distributed according to these constraints (e. g., alignment of spout and cup rim); 3) Common viewpoints, in which these constraints are most evident; 4) Roles of objects and hands, indicating their dominance in the task; and 5) Persistent temporal distribution between events occurring on two arms, indicating bimanual temporal coordination. Leveraging these invariant



(a) Human demonstrations (b) Geometric task representation

Figure 1.2.: Overview of the KVIL approach. (a) Human demonstration videos of manipulation actions involving categorical objects with shape, pose, and trajectory variations aligned in a common viewpoint. (b) Extraction of *sparse keypoints* $\mathbf{k}_1, \mathbf{k}_2$ subject to certain types of *geometric constraints* (point-to-point and point-to-curve), their associated *local frames* $\mathcal{F}$ and the *movement primitives* which represent the demonstrated keypoint motions.

features enhances the generalization capability of VIL across diverse objects and contexts (Kroemer et al., 2021).

A robust robotic VIL framework must incorporate the following key functionalities:

1. A sophisticated visual perception pipeline capable of detecting and tracking relevant object parts and their spatial relationships (attention process).
2. Mathematical formulations of subsymbolic keypoint-based task representations and algorithms for extracting it by leveraging statistical evidence from multiple demonstrations (retention and modeling process). By decomposing tasks into spatial and temporal domains, these algorithms can break down complex problems into smaller, more tractable ones, where motion patterns and constraints become more visible and salient, thereby enhancing robustness and generalization capability.
3. Hierarchical controllers that can reproduce the learned tasks and adapt them to novel objects and scenarios (reproduction process).

The methods developed in this thesis aim improve the task understanding at subsymbolic level, enabling robots to efficiently learn and generalize manipulation skills from visual demonstrations.

The next section formally presents the problem statement that frames the research in this thesis.

1.1. Problem Statement

This thesis aims to develop a robust visual imitation learning (VIL) framework for robotics that automatically extracts subsymbolic task representations from a limited set of human demonstration videos of a sequence of manipulation actions. The proposed framework unifies unimanual and bimanual manipulation tasks by modeling various spatio-temporal bimanual coordination strategies, encompassing the complete bimanual manipulation taxonomy (Krebs and Asfour, 2022). Crucially, the learned task representations should be designed to generalize across intra-category object instances, regardless of variations in object size, shape, and appearance. To this end, we address the following core research questions.

Learning Subsymbolic Task Representations

Object-centric modeling of task constraints, as illustrated in the pouring example (see Figure 1.2b), is key to achieving intra-category generalization. By representing the keypoint-based geometric constraints and their motions in object-centric local frames associated with object functional parts, we can scale and adapt the learned task representation to novel and similar object instances. Here, we explore: 1) What computer vision algorithms facilitate object similarity measurement, invariant feature detection and generalizable object representation? 2) How to model and structure the keypoint-based task constraints based on invariant task features to maximize generalization capability? 3) What statistical algorithms can efficiently extract these invariances, mimicking infant learning processes? and 4) What is the appropriate way to combine adaptable motion representations with task constraints in compliance control framework?

Learning Bimanual Spatial Coordination

Bimanual manipulation, essential in human activities, presents unique challenges due to its complexity involving multiple objects, intricate object relationships, fine-grained motion details, and diverse coordination strategies between both arms. Similarly to the unimanual case, bimanual manipulation tasks also exhibit invariant features over several demonstrations (Muhlig et al., 2009b,a). In this thesis, we investigate: 1) How to extend unimanual task representations to bimanual cases while retaining their properties? 2) What statistical evidence can be used to extract bimanual spatial coordination strategies from sparse data?

and 3) How to control robotic arms to fulfill different coordination strategies and task constraints while maintaining demonstrated motion styles?

Motion Segmentation and Bimanual Temporal Coordination

Human demonstrators often present several sub-tasks in a single demonstration (Wang et al., 2023a) without providing linguistic descriptions of where to segment. The superior cognitive capability allows human to properly segment actions based on contextual evolution and motion characteristics from visual signals. In this thesis, we explore: 1) Which motion segmentation algorithm can be used to segment a sequence of bimanual actions? and 2) How to model bimanual temporal coordination and align action segments across multiple demonstrations, so that the keypoint-based task constraints can be reliably extracted?

It is important to note that our approach excludes linguistic bootstrapping, aligning with studies on pre-verbal or non-verbal infants' ability to grasp motion patterns, object functionalities, and object spatial relations from visual input alone (Meltzoff, 1995; Waismeyer et al., 2015).

By addressing these research questions, this thesis aims to significantly enhance the robustness and generalization capabilities of visual imitation learning frameworks, thereby advancing the state-of-the-art in robotic learning from human demonstrations.

1.2. Contributions

The core contribution of this thesis is the development of a *Keypoints-based Visual Imitation Learning* (KVIL) system, capable of extracting generalizable subsymbolic task representations. KVIL is a bottom-up approach inspired by the observational learning process exhibited by pre-verbal or non-verbal human infants (Meltzoff et al., 2012; Waismeyer et al., 2015), without relying on common sense knowledge embedded in language foundation models (Yiu et al., 2024). To address the challenges outlined in Section 1.1, we present three major contributions in this thesis, namely, 1) modeling and extraction method of keypoints-based subsymbolic task constraints, 2) learning bimanual spatio-temporal coordination strategies, 3) hierarchical motion segmentation, facilitating temporal analysis and automatically learning of a series of actions, including

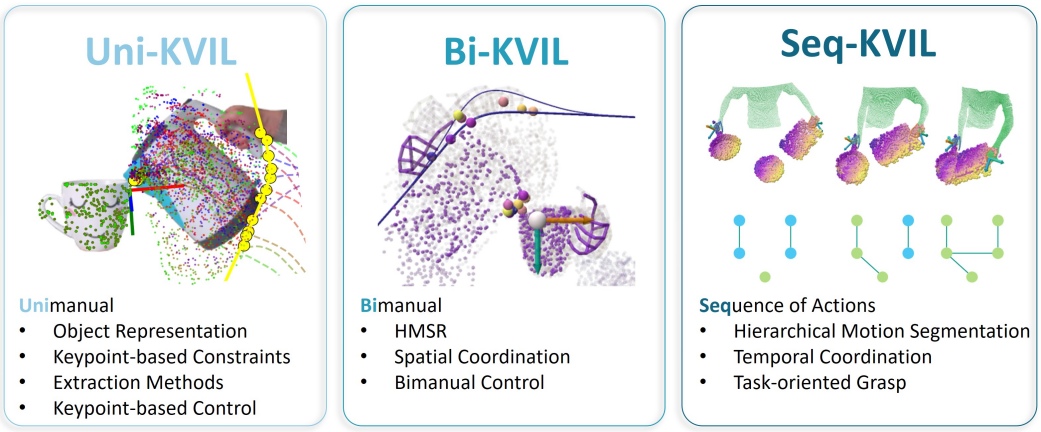


Figure 1.3.: The core contributions of the keypoints-based visual imitation learning approach.

task-oriented grasping. Leveraging multiple types of statistical evidence, KVIL requires only a few visual demonstrations (fewer than 10) to learn effectively. The resulting task representations are sparse, object-centric, viewpoint-invariant, embodiment-independent, and generalizable in intra-category settings. An overview of the contributions is shown in Figure 1.3.

1.2.1. Learning Keypoints-based Subsymbolic Task Constraints

In the first part of the thesis, we propose a visual imitation learning framework that automatically extracts keypoint-based subsymbolic task representations focusing on unimanual tasks, named Uni-KVIL. Key contributions include: 1) A generalizable object representation based on visual neural descriptors in 2D and 3D space; 2) A generalizable subsymbolic task representation comprising keypoint-based geometric constraints on principal manifolds, their associated local frames, and movement primitives for task reproduction; These geometric constraints allow composition for modeling complex spatial constraints. 3) An efficient learning algorithm based on statistical evidence, extracting task representations from a single demonstration video and refining them with additional demonstrations; 4) A novel keypoint-based admittance controller (KAC), which prioritizes geometric constraints, and enables successful task reproduction in novel scenarios.

1.2.2. Learning Bimanual Coordination Strategies

In the second part of the thesis, we extend the keypoints-based visual imitation learning framework to bimanual manipulation tasks, named Bi-KVIL. key contributions are: 1) A *Hybrid Master-Slave Relationship* (HMSR) model structuring the roles and relationships of all object pairs, incorporating Uni-KVIL's task representation for each object pair as necessary; 2) A rule-based system for automatically deriving bimanual coordination strategies from the HMSR graph; 3) A bimanual keypoints-based admittance controller (Bi-KAC), extending KAC, where control commands propagate through the HMSR graph to the robot end-effectors, enabling reproduction of fine-grained bimanual tasks.

Notably, Bi-KVIL simultaneously extracts the HMSR, corresponding bimanual coordination strategies, and keypoints-based subsymbolic task representations from fewer than 10 demonstration videos, while retaining all properties of Uni-KVIL's task representation.

1.2.3. Keypoint-based Segmentation, Bimanual Coordination and Grasping

In the third part of this thesis, we address bimanual temporal coordination by developing an object-centric, keypoint-based hierarchical motion segmentation algorithm. This algorithm produces motion segments with consistent granularity, thereby facilitating the learning of geometric constraints for each segment using Bi-KVIL. The key contributions are as follows: 1) A keypoint-based hierarchical motion segmentation algorithm; 2) A temporal coordination representation for bimanual tasks, which, when combined with Bi-KVIL's spatial coordination, forms comprehensive spatio-temporal coordination strategies for bimanual manipulation; 3) A task-oriented grasp learning, generation and execution framework that leverages the results of motion segmentation. This framework learns task-specific grasp pose constraints in object canonical space for each grasping segment individually, transfers generated target grasp pose to categorical objects, and executes a series of object grasping and rearrangement tasks.

1.3. Structure of the Thesis

The remainder of this thesis is structured as follows. Chapter 2 presents a comprehensive review of the literature relevant to the core problems outlined

in Section 1.1. We begin by discussing structured and unstructured task representations, emphasizing the significance of interpretable intermediate task representations in visual imitation learning. Among structured representations, keypoint-based constraints play a crucial role in enhancing data efficiency and transferability. We analyze various keypoints extraction methods, highlighting their limitations and challenges, and compare them with our proposed approach. Additionally, we review the literature on bimanual coordination and control, as well as motion segmentation approaches for imitation learning.

Chapter 3 introduces the foundational methodologies employed in this thesis. These include state-of-the-art object feature extraction using dense visual descriptors, principal manifold learning algorithms, and movement primitive models. Understanding these components is essential for comprehending the keypoint-based task representations developed in Chapter 4.

In Chapter 4, we detail the *keypoint-based geometric constraints* and propose the *Principal Constraint Estimation algorithm*, which automatically extracts these constraints from the keypoint dataset. We then describe the *keypoint-based compliance controller* for task reproduction, followed by extensive evaluation in various unimanual robot manipulation tasks. To facilitate effective learning from sparse video demonstrations, we propose a neural descriptor-based generalizable object representation and a perception pipeline to obtain dense candidate points, serving as the basis for extracting keypoint-based task constraints.

Chapter 5 extends the keypoint-based task representations and control framework from Chapter 4 to bimanual manipulation. We introduce a *hybrid master-slave object relationship*, which can be automatically extracted from sparse human demonstrations and used to derive the *bimanual coordination strategies* in spatial domain. We further extend the controller to accommodate various bimanual coordination strategies and evaluate its performance across different bimanual manipulation tasks.

In Chapter 6, we introduce a *keypoint-based hierarchical motion segmentation algorithm* that decomposes complete tasks into smaller segments based on keypoint motion characteristics and static and dynamic proximity relationships between object pairs. The segments are then grouped into primitive actions relying on the derived contextual information, facilitating the learning of *bimanual spatio-temporal coordination*. Given the importance of grasping in manipulation task, we introduce *task-oriented grasps* modeled as a pose constraint using Gaussian Mixture Model in object-centric local frames. We then show an application of motion segmentation and task-oriented grasping for object rearrangement tasks.

Finally, Chapter 7 summarizes the core contributions of the thesis. We also discuss "what" is learned as task representations, "how" these representations are derived through invariant features and "when" the spatio-temporal constraints should be extracted. We examine the invariant features that contribute to visual imitation learning and the techniques that facilitate generalization and extrapolation. The chapter concludes by discussing research findings in cognitive development that are yet to be realized in robotics and outlines promising directions for future research in visual imitation learning.

CHAPTER 2

Related Work

The fundamental challenge in imitation learning is to solve the correspondence problem between the demonstrator's and the imitator's context (Argall et al., 2009; Billard et al., 2016; Calinon, 2018; Osa et al., 2018; Liu et al., 2018). This context typically encompasses variations in viewpoints, objects, trajectories of states or state-action pairs, acquired from human demonstrators through various means such as kinesthetic teaching, teleoperation, motion capture, or video recordings using RGB or RGB-D cameras (Argall et al., 2009; Calinon, 2018).

Obtaining data via kinesthetic teaching necessitate direct access to the robot via haptic guidance, which can be challenging to achieve for bimanual tasks (Calinon, 2018). Motion capture systems, though provide precise 3D reconstructions of human motion and object status, demand expensive camera setups and tedious post-processing steps (Argall et al., 2009). In contrast, visual imitation learning offers a more accessible and intuitive approach, relying on video recordings of demonstrations (Kroemer et al., 2021; Liu et al., 2018). This approach significantly reduces hardware costs compared to traditional approaches and enables efficient data collection. However, the reliance on demonstration videos alone introduces new complexities to the context, as it must encompass the appearance of objects and demonstrators, their spatial relationships, and motion trajectories implicitly represented in pixels. A key challenge in visual imitation learning is the absence of direct access to the underlying state of the physical

world and the lack of explicit labels mapping context in human domain to robot's domain (Kroemer et al., 2021). Without predefined action labels, goal states, or action space, visual imitation learning approaches must derive these from raw pixel-level information.

In the literature, the visual imitation learning has been framed as various problems, each addressing a subset of the key challenge: 1) *knowledge retrieval problem* (Pari et al., 2022; Karnan et al., 2022b; Ramachandruni et al., 2020), where the system retrieves stored expert demonstrations that best matches the current perception to guide robot execution; 2) *motion retargeting problem* (Qin et al., 2022), which employs optimization-based techniques to map detected human skeletons in demonstration videos to robot motions; 3) *image and context translation problem* (Liu et al., 2018; Sharma et al., 2019; Smith et al., 2020), where the implicit context information in image space of demonstrator's domain are mapped to the context in robot's domain using deep neural networks, without explicitly extracting explicit intermediate representations; 4) *sequence-to-sequence problem* (Zhu et al., 2023; Fu et al., 2024a), which tokenizes trajectories of contextual information (e. g., object and human hand movements) in the demonstration to train transformer-based policies; and 5) *constraint learning problem* (Jin and Jagersand, 2022; Sieb et al., 2019), where explicit geometric or temporal constraints between objects and demonstrators are modeled at various levels of representation (Kroemer et al., 2021).

Each of these approaches addresses different aspects of the visual imitation learning problem. Knowledge retrieval approaches do not model task constraints, making them dependent on storing all demonstration videos, which can limit transferability, especially in cluttered environments where visual appearance variations affect retrieval accuracy. Motion retargeting approaches primarily focus on mapping kinematic motions between human and robot embodiment. While handcrafted hand-object constraints can be used to design optimization objectives, these approaches often overlook the learning of general task constraints between hands and objects.

Approaches based on image and context translation model unstructured task constraints implicitly using deep neural networks. While these methods can be effective, they require large amounts of training data and are susceptible to viewpoint mismatches. In contrast, structured task representations explicitly model the context using interpretable intermediate representations, such as objects, parts, or keypoints, along with their spatio-temporal constraints from demonstrations. This enables object-centric representation for better generalization compared to unstructured task representation. The constraint learning and

sequence-to-sequence modeling approaches fall within this structured paradigm but differ in their choice of task features and learning methods, leading to variations in data efficiency, transferability, and scalability. The primary challenges in visual imitation learning of structured task representations can be summarized as: (i) Detection of visual correspondences between the demonstrator's and imitator's contexts; (ii) Understanding fine-grained task constraints and scene structures while designing generalizable task representations; (iii) Developing sample-efficient and scalable bimanual coordination strategies and control policies; and (iv) Ensuring efficiency, reusability and transferability in learning long-horizon tasks.

In the following, we review the literature on structured and unstructured task representations in visual imitation learning. In Section 2.1 and Section 2.2, we discuss different task representation models, with a focus on keypoint-based approaches. We discuss how the choice of models and the learning methods affect their generalization capability. Section 2.3 presents methods for modeling bimanual coordination strategies and control in visual imitation learning, while Section 2.4 examines how long-horizon tasks are decomposed into more tractable components through motion segmentation algorithms and bimanual temporal coordination.

2.1. Structured Task Representation

Human infants demonstrate remarkable cognitive capacities in learning and generalizing manipulation skills associated with object classes (DeLoache et al., 2004), and in discovering novel tool usage based on object similarity (Rawlings and Legare, 2021). Experimental evidence suggests that infants can abstract motor skills in an abstracted object space from only one or a few demonstrations (DeLoache et al., 2004), enabling them to project learned skills to visually similar object instances, even under significant variations in object size, properties, and appearance. This cognitive prowess has inspired roboticists to seek similar object and scene representations that facilitate efficient learning of generalizable skill representations. In the literature, the understanding of fine-grained scene structures is primarily achieved through three key aspects: *1*) modeling of object-centric and invariant representation; *2*) modeling of object-centric task constraints on top of this invariant representation; and *3*) extraction of a hierarchy of the scene structure. At the core of these aspects lies the model that measures object similarity at different granularity levels, which is often called

"neural descriptors" in computer vision and robotics (Xie et al., 2022). In the next section, we first review neural descriptor models in Section 2.1.1, then examine each of these key aspects in detail in Sections 2.1.2 to 2.1.5, respectively, where Section 2.1.4 specifically focuses on analyzing different keypoint acquisition methods.

2.1.1. Neural Descriptors

Robot manipulation has benefited from the significant advancements in feature representation and correspondence detection, evolving from traditional handcrafted features to sophisticated neural implicit representations. These features, also called *neural descriptors*, serve as similarity measure at a different granularity, and enable correspondence detection at pixel, part or object instance level (Kroemer et al., 2021).

Early approaches to feature detection relied on handcrafted algorithms such as SIFT (Scale-Invariant Feature Transform) (Lowe, 1999) that were robust to scale, rotation, and illumination variations. While effective for certain visual servoing tasks (Maxim et al., 2012), SIFT had limitations in dense feature extraction and robustness to viewpoint changes. Subsequent work led to the development of dense feature detectors like SuperPoint (DeTone et al., 2018), which leveraged deep learning to improve detection and tracking of salient features across the entire image, facilitating VIL of autonomous navigation tasks following a knowledge retrieval manner (Karnan et al., 2022b). Similarly, general dense keypoint tracking algorithms like SpatialTracker (Xiao et al., 2024) and RoboTAP (Vecerik et al., 2024), demonstrated reliable behavior in tracking local features over time. They perform well in tracking the same objects over time. However, due to lack of consistent semantic meaning in the features, these neural descriptors do not scale to tasks such as finding correspondence between categorical objects (Vecerik et al., 2024).

To enable continuous, fine-grained, category-level neural descriptors, Florence et al. (2018) proposed Dense Object Net (DON), which maps each pixel of an RGB image to a high-dimensional feature space. Pixels with similar local appearance are mapped to nearby locations in this space, enabling *pixel-level dense correspondence detection* based on feature similarity. Unlike sparse keypoint detection algorithms such as SuperPoint, DON's continuous feature space allows arbitrary pixel selection in one image and retrieves corresponding points in another, making it more flexible for fine-grained perception tasks. Trained via

self-supervision, DON is task-agnostic and remains consistent across time, viewpoints, and object instances, facilitating diverse manipulation tasks for both rigid and non-rigid objects (Florence et al., 2018; Hadjivelichkov and Kanoulas, 2022; Florence et al., 2020; Ganapathi et al., 2021; Manuelli et al., 2020). To enhance efficiency and quality in training data collection, we leverage neural radiance fields (NeRF, Müller et al. (2022)) as a more advanced alternative to traditional 3D reconstruction methods used in the original DON approach. However, DON is typically trained on a per-category basis or across a limited set of object categories, making its extension to an open-world setting challenging.

Recent advancements in large visual foundation models have further revolutionized feature extraction for object representation and semantic understanding (Firoozi et al., 2025). Models such as DINO-ViT (self-**DI**stillation with **NO** labels - Vision Transformer, Amir et al. (2022)), DINOv2 (Oquab et al., 2024), SAM (Segment Anything Model, Kirillov et al. (2023)), CLIP (Contrastive Language Image Pre-training, Radford et al. (2021)), Stable Diffusion (Rombach et al., 2022; Tang et al., 2023), and Depth Anything (Yang et al., 2024), RADIO (Ranzinger et al., 2024) have demonstrated remarkable capabilities in extracting rich, semantically meaningful, and/or spatial features from images. These models, pretrained on vast and diverse image datasets, offer powerful visual representations that can be leveraged for various downstream tasks without domain-specific fine-tuning or retraining.

DINO models provide universal feature representations across a wide range of tasks, from pixel-level to image-level. They embed semantic information, enabling point-, part- and instance-level correspondence detection. Wang et al. (2023b); Ju et al. (2024) and Tsagkas et al. (2024) used DINO features at point-level to transfer grasp position on different object instances, while Lin et al. (2024) used it as attention mechanism to train Behavior Cloning (BC) policy. At a part-level, DINO features can be combined with different clustering methods to extract a sparse set of keypoints on the object, which can then be used by optimization framework (Huang et al., 2024b), transformer policy (Di Palo and Johns, 2024b), or graph neural network (Vosylius and Johns, 2023) to generate generalizable robot actions. At object instance-level, Di Palo and Johns (2024b) proposed to use the DINO features for matching object instances between demonstration and deployment, which is then used by a transformer policy to generate actions. Di Palo and Johns (2024a) proposed an imitation learning approach leveraging DINO features at instance-level for trajectory retrieval and at point-level for end-effector alignment with the demonstration.

The quality and expressiveness of these features can be enhanced by merging features from multiple visual foundation models. Tsagkas et al. (2024) demonstrated this by combining DINO and Stable Diffusion features, joining their strengths in semantic understanding and geometric encoding to distinguish semantically similar but spatially distinct object parts for grasping tasks (see also Zhang et al. (2023)). Similarly, Huang et al. (2024b) explored various combinations of DINO, SAM and CLIP features, showing that DINOv2 provides sharper semantic regions compared to ViT and CLIP, while SAM excels at segmenting objects from the background. To further improve the granularity and multi-view consistency, these features can be distilled from multiple images into 3D scene representations. Feature Fields for Robotic Manipulation (F3RM, Shen et al. (2023)) achieves this by distilling DINO or MaskCLIP (Zhou et al., 2022) features into a neural field, generating 3D descriptors for arbitrary interaction point around objects. Similar approaches have been explored in neural implicit scene representations, including FeatureNerf (Ye et al., 2023), LERF (Kerr et al., 2023), and Feature 3DGS (Zhou et al., 2024). However, these methods require dense views and entail time-consuming re-training for each new scene. To eliminate retraining, Wang et al. (2023b) proposed D3Fields, which fuses DINO features of a sequence of images from four camera views to construct a view- and time-consistent feature field of dynamic scenes. This representation enables extracting and tracking of descriptors or keypoints from demonstrations for robot policy learning. However, multi-view setups are often impractical for humanoid robots, limiting the applicability of the distillation-based approach.

In this thesis, we adopt similar ideas as described in Tsagkas et al. (2024) by combining 2D DINOv2 feature (Oquab et al., 2024) with Stable Diffusion feature (Tang et al., 2023) and lifting them to 3D using point clouds derived from depth images. While this combined feature space generalizes well in open-vocabulary settings – eliminating the need to train a DON model per object category – it lacks the precise pixel-wise dense correspondence detection achieved by DON. To bridge this gap, we propose integrating a smoothing deformation field based on the Thin-plate-spline (Duchon, 1977) technique, using the combined feature space as a guide for deformation (see details in Section 4.1.1), which is further used in keypoint extraction algorithm (see Section 4.2).

While fusing or distilling 2D features into 3D typically requires a multi-view camera setup, another research direction focuses on learning 3D features directly from object 3D data, such as point clouds or triangular meshes. Wang et al. (2019) and Wen et al. (2022) proposed normalizing different object instances into a unit cube and finding correspondences based on their coordinates in

this canonical space. However, ignoring the varying scaling factors of different semantic object parts can lead to incorrect correspondence, particularly in the case of significant local shape variation. To address this, methods such as Neural Descriptor Fields (NDFs, Simeonov et al., 2022a) and Neural Implicit Feature Transform (NIFT, Huang et al., 2023)) encode local geometric information within neural implicit representations, learned in a self-supervised manner for 3D shape reconstruction. These approaches provide view-invariant features, enabling robust correspondence detection, similarity measurement, and category-level pick-and-place tasks. However, their dense features struggle to capture fine-grained details in certain object categories. To overcome this limitation, we propose a novel, more informative 3D neural descriptor model, *Multi-feature Implicit Model* (MIMO, see Section 4.1.2), which enhances fine-grained feature distinction and is later applied in Section 6.3 for task-oriented grasping. MIMO outperforms the state-of-the-art models such as NDF and NIFT across various manipulation tasks. However, like NDF and NIFT, these models are typically trained on one or a few object categories, limiting their applicability in open-vocabulary manipulation tasks.

Contributions

In this thesis, we aim to establish a generalizable object representation leveraging neural descriptors. Specifically, we construct object canonical spaces, establish dense correspondences, detect and transfer keypoints, and model task-oriented grasps. To this end, we 1) enhance the data collection pipeline of DON using neural radiance fields (see Section 3.1.1); 2) introduce a novel, more informative 3D neural descriptor model (Section 4.1.2); and 3) integrate a Thin-plate-spline (TPS) deformation field with foundation model features for more reliable dense correspondence detection in open-vocabulary settings (Section 4.1.1).

Leveraging the dense and sparse features as semantic or spatial object descriptors, object-centric representation can be approached in multiple directions, enabling fine-grained manipulation task-representation invariant of camera viewpoints. The following section reviews the state-of-the-art in this direction.

2.1.2. Object-centric and Invariant Representation

Visual imitation learning in robotics seeks to acquire robust object manipulation skills by learning representations that emphasize task-relevant features while discounting distracting, irrelevant details. In many previous works, the desired

motion profiles (Ureche et al., 2015) required for manipulation tasks are conditioned on task parameters (Calinon et al., 2014; Zhou et al., 2020) that describe various scene attributes – ranging from dense visual features, sparse keypoints, and semantic regions to object instances. Exploiting the compositional structure of visual scenes not only emphasizes the features most pertinent to the task but also improves spatial invariance with respect to viewpoint changes, object transformations, and variations in object shape. In this context, constructing an effective object-centric representation necessitates the detection of invariant, task-specific features.

Existing literature has approached this challenge through four main methodologies: 1) contrastive learning to ignore irrelevant features; 2) task-agnostic features; 3) semantic or spatial informative descriptors; or 4) category-level object poses.

Contrastive Learning: Sermanet et al. (2018) introduced Time-Contrastive Networks (TCN), a self-supervised framework that leverages the temporal ordering of video frames to learn discriminative features. By minimizing the feature distance between temporally proximate frames and maximizing it for temporally distant ones, TCN facilitates the learning of viewpoint-invariant latent representations when demonstrations from multiple viewpoints are provided. Nevertheless, this method relies on numerous demonstrations and robot play videos to effectively establish correspondence between human and robot contexts, rendering it embodiment-dependent. Furthermore, when trained on demonstrations of a single task, TCN may inadequately capture fine-grained spatial details and struggle to generalize invariant features across a wider range of tasks.

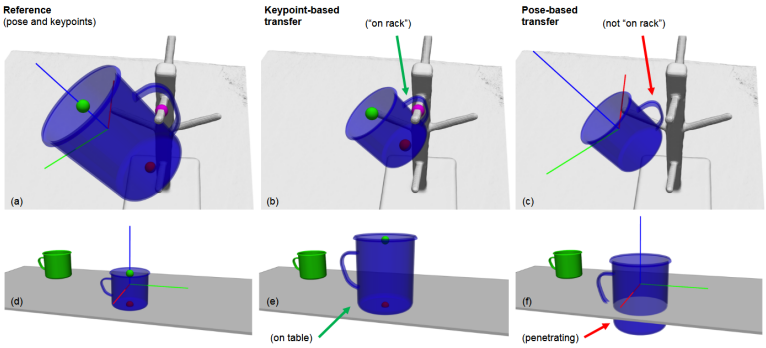
Task-agnostic Features: To improve the data efficiency and transferability, Karnan et al. (2022b) proposed leveraging task-agnostic keypoint detection algorithms – originally applied to vehicle navigation tasks. This approach requires storing demonstration videos and retrieving the most similar demonstration image (knowledge retrieval) based on the maximum number of matching keypoints for reward construction. This reduces the number of demonstrations required by the pixel-level context translators (Smith et al., 2020; Liu et al., 2018; Dwibedi et al., 2018; Karnan et al., 2022a). Similarly, Vecerik et al. (2024) demonstrated that effective robot policies can be learned from as few as 4-6 demonstration videos. However, the task-agnostic nature of these features results in a lack of consistent semantic meaning, as discussed in Section 2.1.1, which complicates the extraction of reliable geometric constraints. Consequently, these methods often average spatial constraints across multiple demonstrations. Moreover, Yang et al. (2022) proposed a transporter-based representation learning model

that extracts keypoints from task-agnostic human and robot play data. However, its reliance on robot execution videos with views matching the demonstrations limits its ability to learn from human demonstrations captured under diverse viewpoints. This requirement is also presented in approaches by Pari et al. (2022); Torabi et al. (2019b,a).

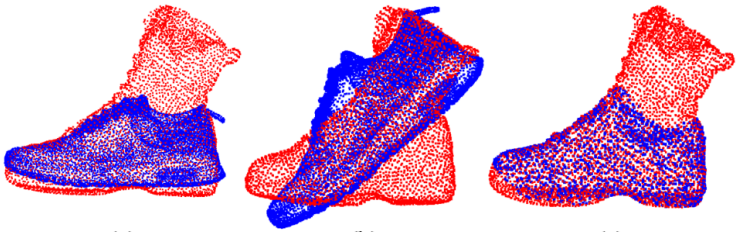
Semantic or Spatial Informative Descriptor: In contrast, approaches that utilize semantic keypoints tied to an object's functional parts have demonstrated superior generalization capabilities and data efficiency. Neural descriptors, such as DON and NDF, capture fine-grained scene features and establish dense correspondences between categorical objects, thereby enabling point-based representation of functional parts. However, approaches by Florence et al. (2020); Simeonov et al. (2022a); Pathak et al. (2018) require access to the robot state space in addition to visual demonstrations, contradicting the purpose of pure visual imitation. Recent advances in visual foundation models have shown promise in representing objects at various granularities, including point-level (Wang et al., 2023b; Ju et al., 2024; Tsagkas et al., 2024), part-level (Huang et al., 2024b; Di Palo and Johns, 2024b; Vosylius and Johns, 2023), instance-level Zhu et al. (2023), and mixed-levels Di Palo and Johns (2024a). This allows downstream robotic manipulation tasks in various granularity and in an open-vocabulary setting.

Category-level Object Poses: Alternatively, object poses and bounding boxes are often used to represent objects at the instance level, though they tend to overlook fine-grained spatial and appearance details of objects. Early imitation learning works leverage instance-level poses as predefined task parameters for motion generation (Calinon et al., 2014; Sena et al., 2019; Huang et al., 2018; Calinon, 2016; Zhou et al., 2020). These works typically assume the known local frames and focus on motion generation to novel object configuration. To derive task parameters from human demonstrations, Perez-D'Arpino and Shah (2017) and Yuan et al. (2018) proposed learning task space regions (TSR) as constraints from teleoperation or kinesthetic teaching data. TSR defines a deterministic region in SE(3) space via a task space pose and a pose deviation margin, which can be computed from multiple demonstrations by identifying the bounds of observed poses. These regions are then incorporated into constraint optimization frameworks for motion planning. However, TSR-based methods require direct robot involvement during demonstrations to capture end-effector poses.

To circumvent this limitation, recent studies have focused on estimating category-level object poses from image inputs (Guan et al., 2024). Such approaches enable the automatic extraction of object 3D poses (or positions) from demonstration videos, which can subsequently be used for waypoint modeling (Jonnavittula



(a) Pose versus keypoints for hanging and placement tasks.



(b) Rigid and non-rigid registration.

Note: Reprinted from Manuelli et al. (2019). © 2019 Springer.

Figure 2.1.: Problems with category-level pose representation.

et al., 2025), keypoint-to-pose mapping (Sundaresan et al., 2023), or the formulation of reward functions for reinforcement learning policies (Patel et al., 2022).

Contributions

Although pose-based methods simplify the representation of manipulation tasks, they often lack the spatial and visual detail necessary for fine-grained manipulation tasks (Manuelli et al., 2019). Specifically, category-level pose defined on a fixed location on a template object may yield a physically feasible goal state in some tasks while failing in others. For instance, given a predefined reference pose at the opening of a smaller cup, transferring this to a large cup still yields a successful pouring action. However, for a precise placement task as shown in Figure 2.1a, this leads to an infeasible target pose. Additionally, traditional pose estimation techniques – whether based on rigid or non-rigid registration – often fail with substantial shape variations among objects, thereby limiting their generalizability (see Figure 2.1b).

In summary, contrastive learning approaches are inherently data-intensive, posing a challenge for tasks where learning must be achieved from sparse demon-

strations. Category-level object poses, by globally representing entire object instances, neglect the contribution of individual functional parts and their shape variations, reducing their suitability for generalizable task representations. Similarly, task-agnostic features, while efficient, suffer from a lack of semantic consistency and category-level generalization. In contrast, semantic and spatial informative neural descriptors offer multi-granularity representations that facilitate learning fine-grained task details from sparse data. Moreover, these descriptors are not limited to rigid and familiar objects, thereby extending their applicability to a wider array of manipulation tasks (Huang et al., 2024b). Based on these considerations, we adopt semantic and spatial informative neural descriptors to form the object representation in Section 4.1. The next section details the various constraints typically modeled on top of this generalizable object representation.

2.1.3. Keypoint-based Constraints

Keypoint-based object representations offer remarkable flexibility for robot manipulation tasks. A single keypoint can denote a grasping location (Tsagkas et al., 2024) or highlight a semantic region corresponding to an object's functional part (Oquab et al., 2024); Multiple keypoints may collectively define the local pose of an object part (Simeonov et al., 2022a), or even form a special geometric primitive, such as a line, plane, or curve, that characterizes spatial relations within the scene Jin and Jagersand (2022); When embedded in a neural network, a set of keypoints can encode the latent state of objects by integrating information on the pose, local geometry, and semantics, while also capturing relational cues among multiple object instances (Wen et al., 2024; Di Palo and Johns, 2024b; Chang et al., 2025). This versatility makes keypoint-based imitation learning approaches particularly adaptable for addressing a wide spectrum of manipulation challenges. In the following, we review four types of *keypoint-based constraints* that have either been exploited in visual imitation learning or show promising potential: 1) neural implicit constraint; 2) point set as template; 3) arithmetic operations; 4) keypoint-based geometric constraints.

Neural Implicit Constraint

Human demonstration videos inherently contain rich information regarding motion, semantics, and physical interactions. End-to-end approaches (Liu et al.,

2018; Sharma et al., 2019; Smith et al., 2020) reframed imitation learning as an image or video translation problem between contexts However, such formulations force the model to jointly learn both physical dynamics and visual appearance, which not only leads to computationally intensive training but also increases the risk of generating hallucinated or unrealistic outputs. By contrast, reducing the input from dense, pixel-level data to a sparse set of keypoints allows for a decoupling of physics, appearance, and policy learning. With keypoint sets serving as latent representations of demonstration videos, policy models are relieved from the burden of implicitly extracting appearance features, thereby enhancing sample efficiency.

For example, Wen et al. (2024) introduced the Any-point Trajectory Model (ATM), which predicts future keypoint trajectories from a random set of keypoints located on both the object of interest and the human body, conditioned on linguistic task descriptions. The predicted trajectories subsequently guide the training of a transformer-based policy to generate robot actions. Nonetheless, the training of ATM demands a large corpus of cross-embodiment videos – including an extensive collection of action-less videos from one embodiment and at least ten demonstrations from another. This dependence on specific embodiment and viewpoints restricts its scalability; deploying the policy on a different robot or from an alternate viewpoint necessitates retraining or fine-tuning both ATM and the policy. Furthermore, the inherent redundancy in keypoint selection complicates learning. To mitigate this issue, Wang et al. (2021) proposed a framework that first employs a 3D attention network to propose keypoints, followed by an attention-switch network that selects task-relevant keypoints during different phases. This end-to-end training strategy, however, still requires a substantial amount of domain-specific data.

To enhance transferability across varying object appearances, recent approaches have incorporated general object descriptors from off-the-shelf foundation models. For instance, Di Palo and Johns (2024b) utilized the Best Buddies Nearest Neighbor algorithm to cluster DINO descriptors of an object into a sparse set of keypoints, which are then used to train a transformer policy. To avoid complications related to embodiment mapping, this work relies on demonstrations collected via kinesthetic teaching and models end-effector motion using only three keypoints, thus unifying the motion representations of both the object and the robot within a keypoint-based transformer policy. Similarly, works by Chang et al. (2025) and Lin et al. (2024) employ DINO features to improve the data efficiency of behavior cloning policies, although these works do not directly use visual demonstrations.

Another line of work employed Graph Neural Networks (GNNs) to model keypoint dynamics in 3D space. These dynamic models are integrated into optimization frameworks (Wang et al., 2023b) and optimal control frameworks (Chen et al., 2023) to facilitate efficient task reproduction. By leveraging general object descriptors, such methods have demonstrated improvements in both data efficiency compared to approaches that use unstructured pixel-level data, and transferability over less informative keypoint detectors. Nonetheless, the implicit nature of these constraints – whether represented via keypoint dynamic models or neural network-based policy models – means that they still require a substantially larger number of training samples than would be necessary for human learning.

Point Set as Template

Rather than implicitly encoding task constraints within neural networks, one can directly use the demonstrated keypoint set as a template. During deployment, the system minimizes either the Euclidean distance between the test keypoint set and the demonstrated template or their distance in a learned descriptor space.

For example, Simeonov et al. (2022a) employed a Basis Point Set (BPS, Prokudin et al., 2019) as a query set to compute a template in a descriptor space embedded by a Neural Descriptor Field (NDF). Typically, BPS is randomly sampled within a predefined bounding box (e. g., around the tool center point of the end-effector), which is effective for grasping tasks but less so for object rearrangements. To gain finer control over point sampling for object rearrangement, Simeonov et al. (2022b) demonstrated the selection of an interaction point via kinesthetic teaching, while Biza et al. (2023) derived the interaction point from the closest contact between an object pair. In both cases, BPS are sampled around the identified interaction point. Alternatively, Huang et al. (2023) introduced a Neural Interaction Field (NIF) and an Interaction Bisector Surface (IBS) to sample query points along the bisector surface between two object meshes, thereby enhancing precision in rearrangement tasks. For skill generalization at the object-part level, Chun et al. (2023) proposed a local NDF that retains a manipulation strategy akin to that in Simeonov et al. (2022a). Since these methods focus on rigid objects – where a point set determines a unique homogeneous transformation – the resulting template is often referred to as a "pose descriptor". Template matching is accomplished by optimizing the homogeneous transformation of the test keypoint set to minimize the distance between its pose descriptor and

the demonstrated template in descriptor space, after which motion planning guides the system to the optimized target pose.

To additionally replicate demonstrated motion patterns, Vecerik et al. (2024) extracted dense keypoint trajectories from demonstrations as references and employed a visual servoing system to continuously align the test keypoint set with the demonstrated pattern over time, culminating in a match with the final goal template. Collectively, these studies have demonstrated the feasibility of one- and few-shot imitation learning in manipulation tasks.

Furthermore, to represent more complex constraints beyond a single object pair, Sieb et al. (2019) proposed Visual Entity Graphs (VEGs) based on Dense Object Nets (DON). VEGs disentangle scene structure into multiple levels (objects, parts, and points) by sampling a random point set within each object and modeling both the links between individual points and the object center, as well as the links among object centers. This hierarchical approach simultaneously captures object spatial transformations and inter-object relations. A path integral policy was then trained using a similarity loss based on the distance between the point sets of the demonstrator's and the robot's VEGs.

Nevertheless, these methods often contend with challenges such as unstructured, unprioritized keypoints and the absence of an explicit constraint model. This increases the complexity of learning generalizable skills, and the template matching process may result in local minima or incur long optimization times (see Section 6.3 for more detailed discussion).Moreover, when faced with substantial shape variations or large pose discrepancies in demonstrations, the optimization process tends to average across demonstrations, potentially overlooking critical task-specific details.

Arithmetic Operations

An alternative strategy eschews the use of an unstructured dense point set in favor of explicitly integrating geometric constraints via arithmetic operations on keypoints. Huang et al. (2024b) illustrated this approach by clustering DINO features into a sparse set of keypoint candidates, which are then annotated on 2D images. These annotated keypoints, together with a language prompt, are used to instruct a large vision-language model (VLM, e. g., GPT-4o (Achiam et al., 2023)) in generating multi-stage plans. In each stage, goal constraints are represented as a Python program that performs arithmetic operations – such as computing the Euclidean L2-distance or the dot product of keypoint

vectors in a global frame – on a selected subset of keypoints. These operations define spatial relationships between keypoints on the same or different objects and are subsequently employed within a trajectory optimization framework to reproduce robot actions.

Following a similar concept, Huang et al. (2024a) proposed partitioning the scene image into distinct regions using image segmentation and leveraging the common-sense knowledge of a large language model to identify task-relevant regions based on a task prompt. The selected regions are then fitted to line and surface primitives, enabling the extraction of a directed point (e. g., the center of a surface along with its outward normal direction). In contrast to Huang et al. (2024b), this method annotates both points and directional vectors on the input image, which are then provided to a VLM to generate task-specific arithmetic constraints, such as point-to-point or line-to-line alignments, parallelism, perpendicularity, and metric distances along a given direction. These constraints are subsequently used in a constrained motion planning framework to achieve the desired robot action.

Both approaches utilize large vision foundation models to propose keypoints and annotate RGB images as visual prompts, which, when combined with linguistic prompts, yield arithmetic constraints via VLMs. Although these methods have achieved promising results in open-world generalization tasks, they primarily follow a top-down visual manipulation paradigm rather than focusing on visual imitation learning. Their reliance on the common-sense knowledge inherent in VLMs and large language models stands in contrast to the goal of this thesis – that is, *to enable robots to learn skills from a small number of human demonstration videos through a bottom-up approach, without the need for extensive linguistic bootstrapping*. Moreover, while these methods allow robots to perform a wide array of open-world tasks, they tend to generate common-sense motions that lack the personalized motion styles found in human demonstrations.

In this thesis, we advocate a bottom-up approach that captures the fine-grained motion style of the demonstrator, thereby mirroring the human observational learning process, and facilitating personalization as a potential future research direction.

Keypoint-based Geometric Constraints

In the previous approaches – neural implicit constraints and point set templates – the geometric constraints are encoded implicitly within neural networks or

templates, without a formal, explicit geometric formulation. Although arithmetic operations can yield explicit constraint formats, their expressiveness is often insufficient for modeling more complex relationships.

Hand-crafted geometric constraint: Explicitly modeling (keypoint-based) geometric constraints has a long history in robotics. Early visual servoing methods (Dodds et al., 1999; Hespanha et al., 1999; Gridseth et al., 2016) manually defined task constraints as simple geometric relationships (e. g., point-to-point or point-to-line constraints). However, due to the limited capabilities of early computer vision systems, these hand-crafted constraints could not be robustly generalized to categorical objects.

Category-level keypoints: Advances in deep learning have enabled the detection of *category-level keypoints* that improve the adaptability of constraint-based imitation learning. For example, Manuelli et al. (2019) introduced kPAM, a keypoint detection model trained on large annotated datasets to detect a fixed set of keypoints for each object category, with each keypoint corresponding to a functional part. For instance, cup has three keypoints at the center of its opening, handle and bottom. In a manipulation task, task-specific alignments, such as aligning a kettle's spout with a cup's opening, are manually defined between these keypoints and then used in motion planning. Subsequent work has enhanced these representations by incorporating object shape completion for collision avoidance (Gao and Tedrake, 2021b) and by augmenting keypoints with orientation information for improved control (Gao and Tedrake, 2021a). However, because the augmented orientations are manually specified and the keypoints remain task-agnostic. For instance, the three keypoints of a cup remain the same regardless of how the cup is used in different tasks, which contradict our intuition as not all keypoints are needed for a placement task. Thereby, these methods limit the discovery of novel tool-use strategies, i. e., the discovery of keypoints on tools based on demonstrations, especially for part of the object that has not yet been labeled. Moreover, they require considerable manual effort to define constraints for different tasks.

Task-specific keypoints: Visual imitation learning (VIL) aims not only to reproduce an action under goal-directed constraints but also to learn such constraints directly from human demonstrations, thereby reducing manual labeling. These task-specific constraints can be modeled via either a single keypoint or a set of keypoints.

A single point: Researchers have demonstrated that a single demonstration of a task-specific keypoint – such as a grasp or interaction point – can be transferred

to novel scenes via neural descriptors. For instance, Florence et al. (2018) used DON to map a demonstrated grasp point to new instances within a category, and Ganapathi et al. (2021) extended this idea to transfer interaction points on fabrics. Owing to challenges in training DON for open-world settings, Tsagkas et al. (2024) leveraged pre-trained neural descriptor models (e. g., DINO and Stable Diffusion) to achieve effective grasp point transfer. Unlike category-level keypoints, these task-specific keypoints can be arbitrarily defined on objects; however, because a single point does not fully constrain a manipulation task in 3D space, additional processes, such as grasp synthesis and motion planning, are required.

A set of point: For more complex tasks, a single keypoint is insufficient to encapsulate the necessary constraints. Jin and Jagersand (2022); Jin et al. (2020a,b) proposed a framework in which geometric constraints are represented as multi-entity relationships among sets of feature points, parameterized by an undirected graph. In this framework, geometric primitives (e. g., point-to-point, point-to-line, line-to-line, and point-to-corner relationships) are defined from detectable keypoints on a static image. For each primitive type, a corresponding Visual Geometric Skill (VGS) kernel is implemented via a graph neural network. The process involves first extracting a dense set of candidate keypoints on the object surface, then exhaustively enumerating all potential point combinations that may form these geometric primitives, and finally selecting those combinations that satisfy the VGS kernels. As a result, a set of point combinations, with each representing a primitive constraint between two objects are extracted from visual demonstration. Although this approach has demonstrated promising performance, several limitations persist: 1) The exhaustive search coupled with graph neural networks is computationally demanding and typically requires a relatively large number of demonstrations (around 30). As the combination possibilities scale linearly with the number of points needed to form the constraints, this approach also suffers from the curse of dimensionality for more complex constraints, such as those involving extended curves. 2) Reliance on semantically limited feature detectors forces demonstrations and reproductions to be captured from a consistent camera viewpoint. 3) The approach has been evaluated primarily on a hammering task, with generalization tested only on hammers nearly identical in appearance to those used in training. Therefore, its generalization capability in scenarios with large object shape and appearance variations is unproven. 4) The approach derives constraints solely from keypoints in one static image, limiting its ability to model constraints that manifest across multiple demonstrations.

Manifold: An alternative to VGS kernels is to model geometric constraints using explicit formulations similar to those used in early visual servoing (Dodds et al., 1999; Hespanha et al., 1999; Gridseth et al., 2016). In daily life, we intuitively use points, lines, and planes as basic linear constraints. Learning such representations can be viewed as a parameterization (or embedding) task where these low-dimensional elements in 3D space are extracted via linear functions. For example, Principal Component Analysis (PCA, Pearson, 1901) can extract linear primitives from a 3D point set by identifying its principal components. For nonlinear constraints, such as curves and surfaces, methods including ISOMAP (Tenenbaum et al., 2000), locally linear embedding (Roweis and Saul, 2000), Laplacian Eigenmaps (Belkin and Niyogi, 2003), parallel transport unfolding (Budninskiy et al., 2019), and manifold flow (Cunningham et al., 2022) have been explored. However, these techniques generally require a large amount of data to achieve reliable parameterization. Similarly, Sutanto et al. (2021) proposed a manifold learning method that unifies linear and nonlinear equality constraints as the zero-level set of a neural network, but this approach demands tens of thousands of samples and off-manifold negative samples, which is rarely feasible in robotic imitation learning tasks.

Principal Manifold: Building on the concept of manifold learning, another line of research models nonlinear embeddings as principal curves and surfaces, inspired by early works by Einbeck (1989) and Hastie (1984). Traditional principal curve representations, however, are limited by requirements such as arc-length parameterization – which hinders extension to higher dimensions, inapplicability to self-intersecting curves, e. g., a handwritten digit "8", and issues in condition verification (Meng and Eloyan, 2021). To overcome these limitations, Meng and Eloyan (2021) introduced the Principal Manifold Estimation (PME) algorithm, which incorporates a regularity penalty to jointly learn linear and nonlinear principal manifolds within a unified framework. PME generalizes PCA and principal curves by minimizing a least-squares distance term augmented with a total squared curvature penalty. Unlike the other nonlinear manifold learning approaches that only focus on the local vicinity of data space, PME yielding smooth principal manifold which captures the embedding while predicting reliable expansion trend of the principal manifold, an advantegous feature facilitating extrapolation for robotics. Despite the existence of numerous principal manifold models, their application to constraint modeling in robotic tasks remains largely unexplored – a gap this thesis aims to fill, see Section 4.2.

Contributions

The approach proposed in this thesis is conceptually similar to that of Jin and Jagersand (2022) in that it involves: 1) generating a dense set of candidate keypoints, 2) performing an exhaustive search to identify keypoints that satisfy various geometric constraints, and 3) formulating these constraints to guide the manipulation policy.

However, our methodology diverges significantly in several key aspects. First, we leverage general task-agnostic neural descriptors to transfer task-specific keypoints to test scenes, allowing us to reuse the descriptor model in different tasks while capturing fine-grained constraints directly from demonstrations. This results in superior visual generalization. Second, we replace the computationally intensive VGS learning with an efficient principal manifold learning approach, enabling the extraction of task-specific keypoints and constraints from a much smaller number of demonstrations (approximately 3-5 for linear constraints and fewer than 8 for nonlinear cases). Representing constraints via principal manifolds not only enhances extrapolation capabilities in the face of large object shape variations – a property not previously demonstrated, In addition, we use movement primitives for action modeling, thereby capturing the demonstrator's motion style more effectively than traditional visual servoing systems.

2.1.4. Keypoint Extraction Methods

In this section, we review various approaches for keypoint extraction in visual imitation learning and visual manipulation. The literature can be broadly categorized into three types of keypoints: 1) semantically less-informative keypoints, i. e., visually salient keypoints, 2) semantically informative, category-level keypoints, and 3) semantically informative, task-specific keypoints.

As discussed in Section 2.1.3, we favor semantically informative, task-specific keypoints because they offer better flexibility for learning task-specific constraints using a consistent set of task-agnostic neural descriptors across diverse tasks. Depending on the application – whether at a symbolic or subsymbolic level, whether fine- or coarse-grained, whether 2D or 3D – the descriptor models can be interchanged or composed, as explained in Section 2.1.1.

In this thesis, we leverage dense neural descriptors – specifically, DON, MIMO, and a combination of DINOv2 and Stable Diffusion – for subsymbolic-level task representation to capture fine-grained details, while DINOv2 features are also used at the symbolic level to incorporate affordance information.

We have also reviewed various keypoint extraction methods in Section 2.1.3, each with its own motivations. In the following, we compare these methods and explain our rationale by drawing inspiration from infant observational learning studies discussed in Chapter 1.

Hand-crafted, manual annotation: Keypoints can be manually annotated on reference images (Dodds et al., 1999; Hespanha et al., 1999; Gridseth et al., 2016; Florence et al., 2018; Ganapathi et al., 2021; Hadjivelichkov and Kanoulas, 2022; Tsagkas et al., 2024) or recorded in 3D space via kinesthetic teaching (Simeonov et al., 2022b). Although this method is straightforward, it requires additional manual effort and often direct access to the robot, making it less desirable for our VIL framework.

(Semi-)supervised learning: Keypoints may also be detected using deep neural networks trained on large datasets. Supervised methods rely on extensive, manually annotated keypoint datasets (Manuelli et al., 2019; Gao and Tedrake, 2021b,a; Xu et al., 2021), whereas semi-supervised approaches leverage a small labeled dataset supplemented by a large unlabeled corpus through multi-view consistency constraints (Vecerik et al., 2021). Typically, these models detect category-level keypoints on novel images; however, because the set of keypoints remain fixed regardless of task requirements, they do not allow adjustment of keypoint positions to suit different manipulation objectives and thus fall short of our needs.

Random sampling: Another strategy involves randomly sampling keypoints associated with an object without explicitly considering their structural relationships or importance to the task. In these methods, task representations are implicitly learned by downstream dynamic transition or policy models. Examples include approaches that utilize 1) randomly sampled task-agnostic features (Karnan et al., 2022b; Vecerik et al., 2024; Yang et al., 2022; Wen et al., 2024; Vecerik et al., 2024; Chen et al., 2023); 2) informative 2D descriptors (Manuelli et al., 2020; Sieb et al., 2019); or 3) informative 3D descriptor fields (Simeonov et al., 2022a,b; Huang et al., 2023; Chun et al., 2023). Although informative descriptors facilitate the transfer of learned tasks among similar objects or parts, these approaches typically treat every keypoint equally, ignoring their inherent structure, priority and spatial relation for a certain task. For example, keypoints on a kettle's spout should contribute more significantly to a pouring task than those on its base, while the latter are more important for a placement task. Therefore, an additional filtering or selection step is necessary to formulate robust, generalizable task constraints using a more compact set of keypoints.

Clustering: As the number of keypoints increases, the complexity of constraint extraction and policy learning grows due to redundancy. Clustering can reduce this redundancy by aggregating keypoints into a compact set. Clustering may be performed based on image patch positions (Jin and Jagersand, 2022; Jin et al., 2020a,b) or 3D voxels (Wang et al., 2023b). However, clustering solely on spatial proximity can neglect semantic information, even when the underlying descriptors (e. g., DINO) are semantically rich. To preserve semantics, clustering based on descriptor similarity has been employed (Di Palo and Johns, 2024b; Huang et al., 2024b; Chang et al., 2025), and semantic image segmentation has been used to partition an image into meaningful regions from which keypoints are derived (Huang et al., 2024a). We consider such descriptor-based clustering particularly promising for handling object affordances and improving temporal consistency. However, the resulting resolution of keypoints highly depends on the chosen hyperparameter such as number of clusters or cluster size, which needs to be fine-tuned for object of different size.

Object interaction: Studies in infant observational learning suggest that interaction or contact points between objects (or between humans and objects) serve as critical parameters for motor skills (Gergely et al., 2002; Meltzoff, 1995; Gweon and Schulz, 2011). Similarly, robotics research has exploited contact points as keypoints between robot and object (Sundaresan et al., 2023), hand and object (Mendonca et al., 2023), and even between objects (Biza et al., 2023) to parameterize manipulation policies. While effective, a single interaction point may not fully specify a manipulation task. For instance, the inclination of a kettle during pouring cannot be determined solely by a keypoint on its spout and keypoints tied to different parts of the object exhibit different characteristics of the task (Verduyn et al., 2024). We propose that a contact point should be considered only if it exhibits salient invariance across multiple demonstrations.

Large language model: In recent visual manipulation frameworks, large language models (LLMs) are employed to select a representative subset of keypoints or key regions from a dense set of candidates. For example, Huang et al. (2024b) used GPT-4o to select keypoints from clustered candidates, enabling the constraint model for each task phase to focus on a specific subgroup of keypoints. Similarly, Huang et al. (2024a) employed GPT-4V to choose semantic patches from an image and generate constraint formulations based on an associated keypoint and direction. In these cases, candidate keypoints are annotated as visual prompts, and together with detailed linguistic prompts of the task, they guide the vision-language model in keypoint selection and constraint generation. We noticed that, the linguistic prompt contributes more than the common

sense knowledge embedded in the foundation models, as they contain detailed description for each task phase about which part of the object to interact or even to exclude. For example, *"This task requires the initial grasping of the drawer handle, followed by a linear pull along its normal vector"* or *"Grasp the flower by its stem, not the petals"*. We argue that relying solely on foundation models with prompt engineering does little to advance our understanding of the observational learning process. Moreover, these methods depend on extensive linguistic bootstrapping and does not yet show potential in visual imitation, whereas our thesis focuses on a bottom-up approach that learns directly from demonstration videos.

Proposal and selection: Alternatively, an agent can learn to infer keypoints as task parameters in a self-supervised and end-to-end manner through exploration. For instance, Qin et al. (2020) and Wang et al. (2021) proposed frameworks that integrate a keypoint (or attention) proposal network with a selection (or attention switching) network, trained end-to-end with sparse rewards based on task accomplishment. Similarly, Minderer et al. (2020) trained a keypoint detector using a self-supervised image reconstruction loss, which was then incorporated into a model-based inverse reinforcement learning framework (Das et al., 2021). While promising, these methods typically require large datasets and significant computational resources.

Alternatively, some approaches adopt an exhaustive search strategy, selecting keypoints directly from visual demonstrations. For example, Jin and Jagersand (2022); Jin et al. (2020a,b) propose exhaustively searching through all possible keypoint combinations to identify geometric primitives in a demonstration image. As discussed in Section 2.1.3, this approach is computationally intensive due to the combinatorial explosion of candidates and is limited to constraint primitives derivable from keypoints observable in a single static frame. Moreover, constraints that span multiple demonstration frames have not been adequately addressed by this method.

Discussion and summary: Inspired by cognitive development experiments – which suggest that infants leverage *statistical evidence* to efficiently infer task representations from only one or a few demonstrations (see Chapter 1) – we aim to develop a VIL system that mimics this capability, which has not yet been demonstrated in the literature. Our approach involves aligning multiple demonstrations to candidate viewpoints defined by the object's local geometry. Compared to instance-level object pose alignment approaches, local pose alignment generalizes to both rigid and deformable objects. For a given keypoint detected across N demonstrations, we treat the resulting N-point dataset as forming a potential principal manifold. By fitting a principal manifold to these points and analyzing

the inter-demonstration variance, we obtain statistical evidence regarding the keypoint's salience and invariance via fitness scores, i. e., how well the principal manifold explains the data. We then select the most salient keypoints along with their corresponding primitive constraints defined by the principal manifold. In contrast to Jin and Jagersand (2022), our method derives primitive constraints from a single keypoint observed consistently across multiple demonstrations at a specific timestep. Finally, by leveraging task-specific keypoints identified via task-agnostic general descriptors, our approach achieves robust generalization when transferring the keypoints to similar objects. This strategy offers several advantages: 1) Task-agnostic general descriptors enable intra-category generalization. 2) Task-specific keypoints facilitate the modeling of object-centric, viewpoint invariant, fine-grained task constraints that reflect the demonstrator's style. 3) Viewpoint alignment based on object local features unifies the representation for rigid and deformable objects. 4) Defining constraint primitives on principal manifolds enhances extrapolation capabilities. 5) The integration of principal manifold learning with statistical evidence supports data- and computation-efficient learning. Our approach will be detailed in Chapter 4. Building on our discussion of keypoint extraction and task-specific constraint formulation, we now review the literature on hierarchical scene decomposition – a framework that organizes various levels of object representations into multi-level structures to capture both local object details and global spatial relationships essential for complex robotic manipulation tasks.

2.1.5. Hierarchical Scene Decomposition

To better capture the complexity of real-world tasks and enable more flexible task generalization, hierarchical organization and representation of visual scenes are essential to facilitate understanding and manipulating objects at multiple levels of granularity.

Granularity Decomposition: Kroemer et al. (2021) reviewed scene representation from a granularity perspective by decomposing objects into hierarchical layers ranging from individual points to parts and complete object instances. Their review highlighted that different types of task representations often emerge at distinct layers; for example, spatial relationships are typically modeled at the object instance level, whereas keypoints are extracted at the point level. We extend this perspective by proposing that *task representations can, and should, be defined across multiple layers of object representation*. For instance, dense point-level correspondences can be employed to estimate object pose (Simeonov et al.,

2022a; Huang et al., 2023) or object part pose (Chun et al., 2023), thereby enabling the transfer of learned manipulation skills to new object instances. Similarly, spatial relations may be defined at the level of keypoints, object parts, or full object instances, depending on the task's granularity. This multi-layered view of object representation opens up new possibilities for both intracategory and intercategory generalization, a core research question addressed in this thesis.

Hierarchical Object Relationships: The second aspect of scene decomposition focuses on hierarchical object relationships. In everyday activities, especially those requiring bimanual manipulation, multiple objects interact simultaneously. In such contexts, *object-centric task constraints are not solely related to the agent or defined within a single global frame; rather, objects often exhibit mutual spatio-temporal relationships, forming a complex graph of interactions*. For example, one object can serve as a reference frame for another, much like how a human hand (or a robot end-effector) is guided relative to the object it is grasping, or how a non-dominant hand establishes a frame for a dominant hand in bimanual tasks (Guiard, 1987a; Kimmerle et al., 2010). Importantly, any layer of object representation (point, part, or instance-level) can be utilized within this object relationship graph. For example, in the pouring scenario depicted in Figure 1.2, the cup is positioned "above" the table (instance-level), while the rim of the teacup (part-level) defines a local frame for the motion of a keypoint (point-level) representing the spout (part-level), which in turn maintains an "above" spatial relationship relative to the cup rim. While prior works such as Sieb et al. (2019) have used graph representations to learn these relationships implicitly, our approach seeks to explicitly extract task constraints across different levels.

A primary challenge in modeling hierarchical object relationships lies in identifying the appropriate reference object and determining where on that object a local frame should be defined. Previous studies have exploited these relationships to adapt learned motion patterns to novel object configurations (Calinon et al., 2014; Ureche et al., 2015; Zhu et al., 2022; Sena et al., 2019; Huang et al., 2018; Calinon, 2016; Zhou et al., 2020). However, these approaches typically assume predefined local frames at the object level, anchored to fixed locations on the object.

Ureche et al. (2015) demonstrated that by leveraging variance in motion and force profiles across multiple demonstrations of hand-tool-object interactions, robots can dynamically select which predefined local frame to attend to and execute motions within that frame. However, learning the locations of these local frames from demonstrations is not considered in this work. Other studies have employed spatial variance across demonstrations to efficiently choose local frames

from a set of predefined candidates (Muhlig et al., 2009b), or used human feedback to resolve ambiguities among category-level candidate frames (Franzese et al., 2020). Similarly, Kober et al. (2015) selected a frame that exhibits the lowest combination of inter-demonstration variance derivative (over time) and final goal pose variance. Niekum et al. (2012) mapped endpoints of the same motion segments from multiple demonstrations to all frames and clustered the points. They discarded singleton cluster (not enough evidence) and selected the frame with the most consistent evidence from the clusters, either the one where the endpoints form a clear cluster or the one providing the largest cluster.

Representing learned tasks in such locally defined frames has proven beneficial by the aforementioned approaches, for transferring skills between different embodiment and for designing effective control policies. Nevertheless, these approaches rely on manually defined candidates at the object level and, as noted in Section 2.1.2, category-level poses often fail to capture the fine-grained features necessary for dynamically relocating task-specific reference frames when objects are used in varied ways. Indeed, as pointed out by Sharma et al. (2019) and Manuelli et al. (2019), applying a fixed pose target to a scaled instance of an object may lead to physically infeasible or unsuccessful reproduction.

In this thesis, we demonstrate that by combining dense visual descriptors with a variance-based criterion, it is possible to simultaneously and efficiently extract both keypoints and their corresponding local frames at a fine-grained level (see Section 4.2). This integrated approach allows us to define task-specific reference frames that are both flexible and robust, paving the way for more effective and generalizable visual imitation learning.

2.2. Unstructured Task Representation

Unlike structured task representations, which require human-designed biases to organize task information, unstructured task representation seeks to minimize human intervention by adopting end-to-end learning strategies. In these approaches, deep neural networks implicitly encode task context information directly from raw image data, and map observations from the demonstrator's domain to that of the imitator. This data-driven approach eliminates the need for explicitly defined intermediate representations, instead allowing the network to learn relevant features autonomously.

A common strategy in unstructured task representation is context translation (Liu et al., 2018). To this end, deep neural networks are trained to predict

observations in the imitator's (e. g., robot's) context from those captured in the demonstrator's context. For example, given a human demonstration video, these networks generate a corresponding video of a robot performing the same task, thereby reducing visual disparity between different embodiment. To achieve this, various generative models have been employed to ensure cycle consistency between the two contexts, including convolutional neural network-based auto-encoders (Liu et al., 2018), adversarial generative models (GAN, Smith et al., 2020; Karnan et al., 2022a), and conditional diffusion models (Ko et al., 2024). These pixel-level translators are subsequently used to train reinforcement learning (RL) policies by maximizing the similarity between the predicted and actual robot observations.

Despite their promising performance, such models are often computationally expensive and time-consuming to train. To improve training efficiency, Sharma et al. (2019) proposed decomposing the learning model into a task-specific GAN-based goal generator and a task-agnostic control policy. This strategy infers control commands based on generated goal images in the robot's context, reducing training overhead by allowing the control policy to be trained independently and shared across tasks. However, these approaches frequently depend on training data collected from specific robot models and fixed camera viewpoints, which limits their generalizability across different platforms and sensor configurations. Furthermore, the absence of dedicated loss functions to filter out background visual distractors can degrade performance in open-world settings.

An alternative strategy, proposed by Sermanet et al. (2018) and Dwibedi et al. (2018), involves embedding scene representations from both human and robot contexts into a shared latent space. Their time-contrastive networks (TCN) are trained using surrogate metric-learning losses that enforce temporal consistency and view invariance, thereby enabling efficient encoding of latent state variables for policy training. While TCNs enhance the robustness of scene representations and better attend to task-relevant features, they still require robot motion videos of similar tasks, which contradicts the goal of visual imitation learning that seeks to operate without robot-specific data. Moreover, none of these methods can readily learn complex tasks from only a single or a few demonstration videos.

A common limitation of these approaches is that they encode the latent state of the scene implicitly within neural networks. This implicit representation restricts scalability to categorical objects and hinders the explicit extraction of scene structures in terms of objects and their functional parts. As a result,

such methods often struggle with category-level generalization across different viewpoints, object instances, and robot platforms.

In contrast, our proposed approach, KVIL, decomposes the demonstrator's context into object-centric, viewpoint-invariant, and embodiment-independent representations, and organizes them in hierarchical scene structures incorporating multi-granularity levels of object representation. This eliminates the need for explicit context translation, extensive robot performance data, and large volumes of human demonstration data during the learning phase. By leveraging task-agnostic, pre-trained neural descriptor models that can be shared across tasks, KVIL's representations can be acquired from a single or a few demonstrations, drastically enhancing data efficiency.

2.3. Bimanual Spatial Coordination

Learning fine-grained bimanual tasks from visual observation of human demonstrations has long been a challenging goal in robotics. Such tasks inherently involve understanding complex object relationships, bimanual control, and spatio-temporal coordination, making them considerably more challenging than the simple sum of two unimanual tasks. While most previous studies have addressed isolated aspects of bimanual visual imitation learning, our work integrates multiple ideas into a unified framework that encompasses both uni- and bimanual manipulation tasks. In this section, we review the literature concerning the representation of complex object relationships and spatial coordination in bimanual manipulation. For a detailed discussion on temporal decomposition and coordination in uni- and/or bimanual tasks, see Section 2.4.

2.3.1. Object Relationship

Bimanual manipulation tasks often involve the simultaneous handling of two or more than two objects. From an object-centric perspective (see Section 2.1.2), one object usually set a reference frame for the other, and sometimes multiple objects may provide simultaneous frames of reference. If we treat human hands as a special object type, all objects in the scene form a complex graph of interactions (Section 2.1.5). Therefore, understanding the roles and relationships between objects (and human hands) is critical for successful task execution.

Early studies primarily focused on hand and arm relationships while largely overlooking the roles played by objects. For instance, the notions of *dominant*

and *non-dominant* hands (or arms) introduced by Guiard (1987a); Kimmerle et al. (2010) describe asymmetrical bimanual tasks, where the non-dominant hand typically stabilizes an object to define a frame of reference for the dominant hand's motion. In robotics, analogous concepts – such as leader-follower (Zhou et al., 2016; Liu et al., 2022), active-passive arms (Suomalainen et al., 2019), and master-slave relationships (Ureche and Billard, 2018) – are used to design control policies in which the follower (or slave, passive) arm operates within a local frame defined by the leader (or master, active) arm. In this thesis, we adopt the master-slave naming convention following Ureche and Billard (2018) and extend it to describe not only arm coordination but also object relationships, treating human hands as a special class of objects. This unified representation captures the *roles*, *relationships*, and *task constraints* relevant to both objects and hands.

Determining the roles of the two arms or hands has been approached in different ways. For example, Guiard (1987b) conceptually defines the master arm as "the one that stabilizes an object", with the slave arm identified by its higher mean velocity. In contrast, Gribovskaya and Billard (2008) defines the master arm as the one that is more restricted in its motion. Meanwhile, Ureche and Billard (2018) suggests establishing the task's reference frame first and then determining the master hand based on the force-motion relationship. However, this approach relies on force information, which is typically unavailable in visual imitation learning. Our experimental observations indicate that in certain bimanual tasks, the master arm may be more dynamically salient during the complete task execution – that is, it can be less limited or even move faster than the slave arm. This finding motivates our exploration of novel criteria for determining master-slave roles (see Section 5.1 for further discussion).

Most of the aforementioned works model a single master-slave relationship between two arms. This is not readily applicable for multiple objects that forms a graph of interactions as described in Section 2.1.5, where an object set a reference for another, which itself set another reference frame for a third object. Though this chained or inter-dependent relationship is implicitly captured by the Visual Entity Graphs proposed by Sieb et al. (2019), it tends to average the spatial constraints when multiple demonstrations are provided, and the subsequent policy learning demands a large amount of data.

In this thesis, we seek to effectively and explicitly extract task constraints across different object representation levels for each object pairs if necessary in the graph. To achieve this, we proposed a hybrid master-slave relationship (MSR) presented in Section 5.1. The bimanual manipulation categories proposed by Krebs and Asfour (2022) of the demonstrated tasks are then derived from the

extracted MSR (see Section 5.2). It is important to note that, this approach unifies the uni- and bimanual manipulation tasks, and captures fine-grained manipulation styles.

2.3.2. Bimanual Spatial Coordination

Many approaches to bimanual manipulation have concentrated on designing controllers based on pre-defined coordination categories (Ajoudani et al., 2014; Savic et al., 2016; Almeida and Karayiannidis, 2019; Mirrazavi Salehian et al., 2018; Gao et al., 2018; Park and Lee, 2015; Lee and Chang, 2015; Amadio et al., 2019) rather than learning coordination strategies directly from demonstrations. These strategies are typically encoded either implicitly within motion trajectories or explicitly as constraints. In the following subsections, we discuss both implicit and explicit spatial coordination approaches.

Implicit Spatial Coordination

Trajectory-based bimanual imitation learning methods focus on capturing the spatio-temporal correlations of bilateral motions using variations of movement primitives Pairet et al. (2019); Franzese et al. (2024); Dong et al. (2022); Knaust and Koert (2021) or Transformer-based models Liu et al. (2022). Although these methods successfully encode coordination implicitly, they tend to overlook the roles played by objects, thereby limiting their generalization capabilities when compared to object-centric VIL approaches (see Section 2.1.2). Similarly, bimanual deep imitation learning methods Zhao et al. (2023); Fu et al. (2024b); Chen et al. (2022); Kataoka et al. (2022); Xie and Chowdhury (2020); Kim et al. (2024, 2021) implicitly embed coordination strategies but require a large number of demonstrations, which are not always available in real-world scenarios. Despite their effective reactive controllers within trained environments, these methods do not explicitly encode coordination strategies and constraints, which restricts their generalizability, especially when only a few demonstrations are provided.

Explicit Spatial Coordination

In explicit coordination approaches, the representation of coordinated behavior is abstracted by analyzing factors such as contact and grasp states, object/hand roles, as well as spatio-temporal and force constraints. For instance, a rule-based

classification proposed by Krebs and Asfour (2022) categorizes bimanual actions according to a predefined manipulation taxonomy. Other studies have focused on object-action relationships (Dreher et al., 2020a) or on learning specific coordination strategies, such as asymmetric tightly-coupled (Ureche and Billard, 2018). In the present work, we concentrate on spatio-temporal constraints because force data, as required by Ureche and Billard (2018), is typically unavailable in demonstration videos. Specifically, we integrate the bimanual coordination categories proposed by Krebs and Asfour (2022) within our MSR representation and control scheme, ensuring each coordination strategy is realized in real-time compliance controllers. Additionally, we eliminate the need for predefined object frames (Ureche and Billard, 2018), by combining the automatic extraction of MSR, bimanual coordination, and object-centric task representations, including the keypoints, their constraints, and the associated local frames. This unified approach facilitates the learning of generalizable, fine-grained bimanual manipulation skills.

2.4. Motion Segmentation and Learning

In the previous sections, we reviewed the literature on learning geometric constraints (Section 2.1) and spatial coordination (Section 2.3), focusing on uni- and bimanual task representations for individual action segments. However, in practice, humans often demonstrate long-horizon tasks as sequences of actions, making visual imitation learning a significantly more complex problem (Zhang et al., 2024a; Prados et al., 2025; Wang et al., 2023a; von Hartz et al., 2024; Kim et al., 2024). Beyond the challenges of translating high-dimensional, continuous human demonstration videos into executable robot policies, an additional difficulty arises from the natural variability in human behavior, both spatially and temporally, even when performing the same action repeatedly (Gutzeit and Kirchner, 2022).

Monolithic imitation learning approaches, which treat demonstrations as unsegmented sequences (Liu et al., 2022), often fail to generalize due to the inherent variability and compositional nature of human behavior (Graybiel, 1998; Pastor et al., 2009). These approaches suffer from several critical limitations: 1) *Limited insight into sub-task structure*: The inability to explicitly segment tasks makes it difficult to diagnose errors or integrate prior knowledge for each task phase. 2) *Temporal and spatial variability*: Subtle differences in timing and hand or object poses across multiple demonstrations lead to misalignment in spatial and tem-

poral domain, hindering the derivation of constraints or robust policies. 3) *Lack of reusability*: Skills learned for one task cannot be easily reused, necessitating repeated learning from scratch.

These limitations underscore the necessity of frameworks that decompose motion into interpretable, reusable components, reflecting the neural mechanisms underlying human motor control (Graybiel, 1998). Motion segmentation offers a biologically inspired solution by decomposing demonstrations into motion primitives – discrete, semantically meaningful units such as reaching, grasping, or placing (Bizzi et al., 1991; Mussa–Ivaldi and Bizzi, 2000; Flash and Hochner, 2005). This decomposition significantly reduces the complexity of the learning problem.

In this section, we first review the literature on motion segmentation algorithms (Section 2.4.1), focusing on the segmentation of long-horizon tasks into interpretable, reusable motion primitives. We categorize the reviewed works based on the granularity of the segmentation, distinguishing between action segmentation and motion segmentation. Action segmentation focuses on partitioning high-level human activities into semantically meaningful segments, while motion segmentation aims to identify subsymbolic motion characteristics and physical events. We further discuss the challenges of bimanual action segmentation and temporal coordination in Section 2.4.2, which are essential for learning complex bimanual tasks. Finally, we review the literature on learning motions in sequential tasks in Section 2.4.3.

2.4.1. Motion Segmentation Algorithms

Temporal segmentation of demonstrations can be performed at multiple granularity levels, such as symbolic-level segmentation with semantic action labels, subsymbolic-level segmentation based on motion characteristics and physical events, or a combination of both. Based on the learning paradigm, motion segmentation approaches can be categorized into (weakly-)supervised, unsupervised, and self-supervised approaches.

Supervised Approaches

Action segmentation approaches (Yang et al., 2023; Xu and Zheng, 2024) partition high-level human activities (e. g., "opening a fridge", "pouring water") into semantically meaningful segments. While useful for recognizing human actions

at semantic level, these approaches often fails to capture the underlying execution details, resulting in fuzzy or context-dependent segmentation boundaries. For example, a "pouring" action is typically treated as a single segment without distinguishing sub-actions like "moving bottle toward cup" "tilting bottle to pour" and "placing bottle back on the table".

Dreher et al. (2020b) proposed a graph neural network classifier trained with supervised learning to classify scene graphs representing spatial relations between object instances. Their approach provides finer granularity than methods proposed by Yang et al. (2023) and Xu and Zheng (2024), yet struggles with misclassifications in phases like lift-place and retreat-approach, even when dynamic spatial relations (e. g., "moving together" and "moving apart") are considered.

Based on our experiments, we identified three major limitations of supervised segmentation approaches: 1) *Contextual labeling dilemma*: The same motion can receive conflicting labels depending on the reference object, e. g., "retreating from object A while approaching object B". 2) *Granularity discrepancy*: Human-labeled actions (e. g., "approach") may consist of smaller motion segments (e. g., "convergent" and "divergent" motion segments) that are overlooked during labeling, thereby polluting the training dataset and introducing inconsistencies in supervised learning. 3) *Limited generalization*: As noted by Meixner et al. (2023), supervised motion segmentation methods fail to generalize effectively to novel scenes or tasks without retraining or fine-tuning, limiting their application in robotics.

Since imitation learning requires precise temporal alignment for task constraint learning and executable motion policy derivation, we exclude purely supervised semantic-level action segmentation approaches in the remaining sections. Instead, inspired by Wächter and Asfour (2015), we leverage motion characteristics while incorporating contextual information, such as object categories and object contact relations, for robust motion segmentation. Similarly to Dreher et al. (2020b), we construct scene graphs using object proximity status and a set of predefined primitive actions rather than spatial relations, allowing motion segmentation to be determined purely by motion characteristics and contextual information without human labeling (see Chapter 6). In the next, we review self-supervised and unsupervised motion segmentation algorithms that provide precise segment points without relying on manual labeling.

Unsupervised Approaches

In contrast to supervised methods that always require manual labor for annotation, unsupervised methods often leverage the trajectory characteristics of sensor signals to determine segment points, without relying on manual labels. Various characteristics have been explored, including velocity, acceleration, force profiles, their specific patterns, and spatial relationships of objects. Based on the segmentation point detection methods, unsupervised methods can be categorized into threshold-based, kinematic similarity-based, clustering-based, probabilistic, and hierarchical approaches, which will be detailed next.

Threshold-based approaches: A common technique is Zero-Velocity Crossing (ZVC, Fod, 2002), which detects segment points when velocity crosses zero. Variants extend this idea to acceleration (Meixner et al., 2023), force (Su et al., 2016), and other sensor cues. Many works that employed ZVC for motion segmentation only consider the translational velocity trajectories in 3D space as three different signals, and segment them separately (Fod, 2002; Wächter and Asfour, 2015; Meixner et al., 2023). This tends to introduce over-segmentation, as the Zero-Velocity Crossing (ZVC) points in one axis may not imply velocities being zero in other axes. Moreover, overlooking the rotational velocity may lead to under-segmentation, as the rotational velocity often contains important information about the task execution. To address this issue, von Hartz et al. (2024) proposed to factorize the state space using directions and magnitude of both translational and rotational velocities and only use the magnitude of them for segmentation. These methods are simple and computationally efficient, but they often require manual tuning of threshold parameters, making them sensitive to noise and task-specific characteristics (Meixner et al., 2023).

Kinematic similarity and homogeneity: Model-based approaches fit time-series data with locally linear (e. g., PCA, HMM, GMM) (Elhamifar and Vidal, 2009; Barbič et al., 2004; Niekum et al., 2012; Lee et al., 2015; Krishnan et al., 2017) or nonlinear models (e. g., Locally weighted regression Calinon et al. (2010)). These methods segment data when the learned local model in previous time windows fail to explain the current period, i. e. when local models transit. Similarly, trajectory similarity metrics, such as Hausdorff distance, Frechet distance, Dynamic Time Warping (Despinoy et al., 2016), curvature (Tapia et al., 2024; Prados et al., 2025) and Levenshtein Distance (Tapia et al., 2024) have been used to determine or refine segment points, by computing the dissimilarity in trajectories. However, homogeneous motion segmentation tends to over-segment or under-segment in many cases, leading to inconsistencies in segmentation levels. For example,

complex velocity profiles can be segmented into multiple smaller segments to optimize local approximation, while peaks in velocity profiles are usually not necessary segment points. On the other hand, segment points at the extrema of the valleys in velocity profile are usually extracted, without respecting the low-velocity phase as by Fod (2002), which is often important for learning motion constraints.

Clustering in spatio-temporal domain: Clustering algorithms, such as DBSCAN, have been employed to identify dense regions of trajectory points (Hachem and Damiani, 2018; Damiani et al., 2018). These methods exploit the high density of points in areas where objects remain stationary for extended periods, using cluster boundaries to infer motion segmentation points. However, such approaches are inherently limited: they primarily distinguish between static and dynamic phases without capturing interactions between objects. Furthermore, their performance is highly sensitive to hyperparameter selection, particularly when data are recorded at varying frequencies, which directly affects the density of trajectory points. Additionally, these methods predominantly focus on spatial clustering and often fail to incorporate temporal patterns effectively.

To address this limitation, Tsai et al. (2019) proposed a spatio-temporal clustering approach, refining segmentation points by detecting local extrema in the distance profile and identifying ZVC points. While this method improves the segmentation by incorporating temporal information, it does not resolve the dependency on hyperparameter tuning or the inability to model object interactions.

An alternative line of research employs clustering techniques such as Gaussian Mixture Models (GMMs) and K-means to refine initial over-segmentations, which can be obtained using heuristic criteria, including changes in movement smoothness (Lioutikov et al., 2015, 2017; Prados et al., 2025) or bell-shaped velocity profiles (Gutzeit, 2022; Gutzeit and Kirchner, 2022). GMMs are then applied to probabilistically blending segment points from different feature dimensions of the motion in temporal domain (Lioutikov et al., 2015, 2017; Prados et al., 2025), while K-means clustering has also been employed for refinement (Gutzeit and Kirchner, 2022). Nevertheless, these methods often suffer from both over-segmentation and under-segmentation within different parts of the same task. Moreover, their sensitivity to clustering parameter – such as Gaussian kernel size and time window length – hinders their robustness and adaptability to new tasks.

Probabilistic approaches: Human demonstrations exhibit variability in motion speed, sensor noise, and differences in execution styles, posing significant challenges for parameter tuning in clustering- and threshold-based segmentation methods. Probabilistic approaches have been introduced to address these challenges, offering advantages such as the ability to model uncertainty, incorporate prior knowledge, and adapt to varying numbers of motion segments.

Hidden Markov Models (HMMs) and their extensions (Kulić et al., 2012; Song et al., 2020; Rozo et al., 2020; Niekum et al., 2012) are widely utilized for motion segmentation due to their capacity to capture temporal dependencies and model stochastic transitions between hidden states (e. g., motion primitives). In these models, each state corresponds to a specific motion segment, and segmentation is derived by decoding the most probable state sequence given the observed data. However, HMM-based approaches typically require either a predefined number of states (Tang et al., 2010) or are sensitive to the training data. These challenges have been reduced by non-parametric extensions, such as Beta-Process Autoregressive HMM models (Niekum et al., 2012), Dirichlet Process HMM models (Krishnan et al., 2017), or Bayesian Information Criterion. These models infer the appropriate state structure and suitable number of states. However, Lioutikov et al. (2015) and Gutzeit and Kirchner (2022) pointed out that these methods still produce inaccurate segmentations in daily manipulation tasks.

Another line of works employ Bayesian non-parametric models to detect change-points in time series data (Truong et al., 2020). Various search methods, such as dynamic programming (Bai and Perron, 2003), sliding window, Pruned Exact Linear Time (Pelt Killick et al. (2012)), have been proposed to optimize the change-point detection process with different constraints. As an application in robotics, Zhang et al. (2019) proposed to use Bayesian change point detection to segment the robot holistic trajectories collected via kinesthetic teaching and fit Gaussian distribution of GMMs to model the motion primitives, which are then clustered into groups based on the Kullback-Leibler (KL) divergence between the mixture models. Similarly to the clustering based approaches, these methods tends to produce inconsistent levels of segmentation as the semantic information is neglected in the optimization process.

Hierarchical approach: The clustering-based approaches described earlier (Gutzeit, 2022; Gutzeit and Kirchner, 2022; Prados et al., 2025) represent bottom-up hierarchical motion segmentation. These methods first generate initial over-segmentation and then refine the results through a merging or grouping phase. In contrast, top-down approaches first identify high-level semantic events, such

as contact occurrences or spatial relationships between objects, and subsequently apply subsymbolic motion segmentation based on motion characteristics.

A notable example is the work of (Wächter and Asfour, 2015), which detects contact events using 2D object segmentation. However, this approach does not clarify how segmentation at multiple levels can be effectively utilized for imitation learning and motion representation, especially when over-segmentation occurs at the subsymbolic level. Furthermore, contact detection based solely on 2D segmentation masks is prone to occlusion, which may lead to inaccurate interpretations of contact in 3D space.

To enhance the reliability of contact detection, force feedback or a combination of position and force signals has been employed (Su et al., 2016). However, such methods present several challenges: 1) Force-based contact detection is difficult to obtain from visual demonstrations. 2) Measuring object-object contact forces is challenging. 3) Certain tasks involve close object proximity without actual contact (e. g., pouring from a height into a cup). Given that this thesis focuses on human demonstration videos in RGB-D format, we employ 2D object detection and lift the detected results to 3D space using the depth image. We integrate a deformation field, i. e. Thin-Plat-Spline, on top of the keypoint tracking results to address the occlusion problem, generating reliable object motion data in 3D, thereby facilitating reliable contact event detection. Moreover, we leverage the semantic object relations and motion characteristics to refine the segmentation points, ensuring that the motion segments are both semantically meaningful and interpretable. Additionally, we further group sub-segments based on a semantic scene graph representation similarly to that proposed by Dreher et al. (2020b).

Discussion

Object Representations: In many motion segmentation approaches, objects in the demonstration are often over-simplified or even ignored. For example, some approaches represent objects as *single points* (Fod, 2002; Wächter and Asfour, 2015; Prados et al., 2025), which inherently neglects rotational motion and can lead to inaccurate segmentation, particularly for tasks like pouring (Verduyn et al., 2024).

To address this issue, researchers have employed *local coordinate frames* attached to rigid objects, human hands, or robot end-effectors (Tsai et al., 2019; Verduyn et al., 2024; Rozo et al., 2020; von Hartz et al., 2024) for motion segmentation. However, this method is effective only for rigid objects, limiting its applicability

to deformable objects. An alternative approach involves using *(axis-aligned) bounding boxes* (Dreher et al., 2020b) to compute semantic spatial relationships, which is the input of a graph neural network classifier for action recognition. Both local frame and bounding boxes do not consider the complex shape of the objects, thus lack of precision in motion segmentation. More recently, Meixner et al. (2023) incorporated full *object meshes* to detect object contact relation. While this method improves accuracy, it relies on expensive motion capture systems and predefined object models. Furthermore, the object mesh is used only for contact detection, while the velocity and acceleration is still point-based.

Since shape completion methods are often imprecise, prone to hallucination, or computationally expensive, we adopt an object representation based on *a set of keypoints*. More specifically, they are densely tracked keypoints on each object with identity correspondences.This approach accommodates both rigid and deformable objects while effectively capturing translational and rotational motion characteristics. Additionally, when applied to contact event detection, it achieves comparable precision to mesh-based algorithms.

Frame Dependency: Most motion segmentation algorithms rely on a single reference frame (Fod, 2002; Wächter and Asfour, 2015; Meixner et al., 2023; Prados et al., 2025; von Hartz et al., 2024). This dependency makes kinematic features, such as velocity along x, y, z axes, susceptible to variations in frame selection, thereby hindering generalization and introducing challenges when viewpoints differ.

Gutzeit and Kirchner (2022) used the demonstrators' local frame to represent the motion of their hands and the manipulated objects. Though dependency on a single global frame selection is revealed, this choice of frame is not always optimal for motion segmentation, as a static object starts to move when the demonstrator moves. In contrast, Tsai et al. (2019) attempted to mitigate the single-global-frame issue by projecting trajectories into multiple random coordinate systems and employing frame-independent motion descriptors, such as Euclidean distance between objects and trajectory variance in the spatio-temporal domain. Similarly, von Hartz et al. (2024) propose to utilize only the magnitude of velocities as cues for motion segmentation, thereby reducing the dependency on frame selection. However, random frame selection by Tsai et al. (2019) does not ensure a coherent representation of object relative motion, particularly when objects exhibit substantial shape variations.

Verduyn et al. (2024) introduced invariant trajectory-shape descriptors based on screw theory to make motion segmentation independent of specific reference

frames. However, this method assumes rigid bodies, limiting its applicability. In contrast, we propose a *local pose estimation* method (see Section 4.1.3), where we estimate a local frame attached to each densely sampled keypoints on the object leveraging neural descriptors. This enables the projection of keypoints motion of any object (including human hands) to any local object parts, creating a viewpoint-invariant motion dataset and suitable for both rigid and deformable objects. We further use frame-independent motion characteristics like the norm of the velocity as one of the cues for our motion segmentation algorithm. This allows fine-grained relative motion analysis considering both human-object and object-object interaction, which is essential for reliable motion segmentation in complex bimanual tasks.

Hyperparameter Tuning: Various hyperparameters, such as time-window (Barbič et al., 2004), thresholds for ZVC (Tsai et al., 2019), Gaussian kernel parameters, and merging time-window in probabilistic models (Lioutikov et al., 2017; Prados et al., 2025), and complexity penalty of Bayesian change point detection Truong et al. (2020), must be fine-tuned for different tasks. Additionally, we observed that probabilistic and Bayesian approaches are more computation-demanding than threshold-based approaches and tend to both under-segment and over-segment different phases of the same task, making it difficult to define hyperparameters.

Therefore, in this thesis, we choose threshold-based segmentation methods leveraging motion characteristics and semantic object relations. To reduce effort in threshold selection, we propose object-invariant distance thresholds, soft-thresholding techniques, and relative velocity thresholds, which can be heuristically defined for various tasks (more details in Section 6.1).

Bottom-up merging: As discussed by Lioutikov et al. (2015, 2017); Meixner et al. (2023); Prados et al. (2025), heuristic- and threshold-based segmentation methods tend to over-segment the motion. Various merging techniques, such as clustering methods (Lioutikov et al., 2015, 2017; Prados et al., 2025; Gutzeit and Kirchner, 2022) have been explored to group over-segmented components to reasonable more complex action segments, while Gutzeit and Kirchner (2022) explored to group the segments hierarchically.

For example, Gutzeit and Kirchner (2022) proposed to use Gaussian means (g-means) algorithm to automatically cluster the building blocks into the most suitable number of clusters based on how good each cluster resembles a Gaussian distribution. To achieve more segmentation accuracy in the context of object manipulation, they cluster based on motion features including the distance of the

human hand to the manipulated object as well as their speed in the demonstrator's local frame. Though the clustering is performed interactively at different levels, the resulting segments are not necessarily consistent corresponding to task execution levels, as no semantic meaning is considered in the clustering process.

Alternatively, Prados et al. (2025) centered Gaussian kernels at each over-segmentation point in time and computed the likelihood of each timestep using the Gaussian Mixture Model. Then they select the time points corresponding to peaks in the probability distribution function, which is then filtered based on user-defined time window threshold. However, clustering in the temporal domain alone overlooks the semantic information of the segments, leading to inconsistent segmentation and hindering the learning of task constraints and bimanual coordination models.

In contrast, Tsai et al. (2019) applies the DBSCAN clustering algorithm to the spatial data of over-segmentation points from both hands – grouping closely located points and filtering out isolated noise – and then determines the final segmentation points by averaging the time steps of points within each cluster. However, this approach is specifically tailored for surgical data where segmentation points of both hands are clustered at once, which is not suitable for everyday bimanual manipulation tasks where different coordination strategies must be modeled.

Similarly to Tsai et al. (2019), we refine proximity-based segmentation points using velocity profiles derived from distance profiles of the closest point pairs (see Section 6.1.1). These proximity states – such as contact, moving closer, moving apart – are then used to establish object semantic relations and construct scene graphs, similarly to that by Dreher et al. (2020b). Unlike clustering-based merging approaches (Lioutikov et al., 2015, 2017; Prados et al., 2025; Gutzeit and Kirchner, 2022), our method merges sub-segments based on contextual information derived from object proximity status and motion characteristics without relying on probabilistic models in temporal domain, thereby avoiding inconsistent levels of segmentation.

Most of the motion segmentation methods are developed for and evaluated on unimanual manipulation tasks, in the next, we focus on bimanual motion segmentation, which is essential for learning complex everyday tasks.

2.4.2. Bimanual Motion Segmentation

Bimanual tasks introduce unique challenges for motion segmentation due to the need for precise spatio-temporal coordination. Each arm must execute distinct yet interdependent movement primitives (e. g., one stabilizes an object while the other manipulates it). Proper segmentation allows for the explicit modeling of coordination constraints and facilitates the analysis of inter-arm synchronization.

Tsai et al. (2019) segmented 6-DoF trajectories of both hands, detecting translational and rotational segment points individually using local extrema of distance profiles and ZVC points. Segments were subsequently merged by clustering poses of both hands in the spatial domain without distinguishing each arm individually. Their approach primarily addressed bimanual surgical motion and did not account for more complex coordination strategies or cases where the two hands operate independently.

Gutzeit (2022) introduced a hierarchical approach, first segmenting motions of each arm into building blocks followed by a hierarchical merging step that groups segments into more complex actions at different levels. However, their method focused only on tasks where both arms started and ended simultaneously, or cases where only one arm is moving, neglecting more intricate temporal coordination strategies.

Motion segmentation algorithms based on supervised learning (Dreher et al., 2020b) do provide semantic action segmentation but necessitate manual labeling and training or fine-tuning data for new tasks, thereby not readily applicable to visual imitation learning with sparse dataset.

Most existing unsupervised bimanual segmentation approaches prioritize motion cues from the dominant hand or jointly segment both hands' movements without explicitly accounting for different coordination types. Consequently, their segmentation results are insufficient for learning task constraints and modeling bimanual coordination in both spatial and temporal domains.

2.4.3. Motion Learning for Sequential Tasks

Some works integrate motion segmentation into motion primitive learning frameworks like Task-Parameterized Gaussian Mixture Models (TP-GMMs, Calinon, 2016). For instance, von Hartz et al. (2024) proposed to segment motions using ZVC on the magnitude of the translational and rotational velocity

and then determine the segmentation points as the temporal center of the filtered segments. Then they align motion segments from multiple demonstrations to model the motion using TP-GMMs. The selection of task parameters, typically the local frames of reference tied to different objects, is similar to our approach. They first sample candidates within an object, using neural descriptors to establish correspondence between objects, and then model the relevancy score of each candidate over time using Gaussian precision matrices. The most relevant candidate is selected as the local frame of reference for that motion segment. However, a single local frame per motion segment is not readily sufficient for complex bimanual tasks, where multiple local frames are required to model the spatial constraints between multiple objects and hands. On the other hand, though the objects are used for the selection of local frames, their trajectories are ignored for motion segmentation, which we believe is important for object-centric task representations. Moreover, this work focuses primarily on modeling the motions while overlooking the motion constraints, which is essential for learning the task structure and generalizing to novel scenarios. Similarly, Mendez et al. (2024) refined TP-GMM segmentation using Kullback-Leibler divergence. However, TP-GMM-based methods rely on predefined movement libraries, which limit their generalizability to novel tasks.

2.4.4. Contributions

In this chapter, we have reviewed the literature on motion segmentation algorithms, focusing on their application to visual imitation learning for long-horizon tasks. We categorized the approaches into supervised and unsupervised methods, highlighting their respective strengths and limitations.

Supervised methods, while effective in providing semantic action segmentation, require extensive manual labeling and struggle with generalization to novel tasks. Unsupervised methods, on the other hand, leverage trajectory characteristics and clustering techniques to identify segment points without labeled data. However, they often suffer from over-segmentation and under-segmentation issues, and their performance is highly sensitive to hyperparameter selection. Probabilistic approaches, such as Hidden Markov Models and Bayesian change point detection, offer advantages in modeling uncertainty and incorporating prior knowledge but are computationally demanding and prone to inconsistent segmentation levels.

We also discussed the challenges specific to bimanual motion segmentation, emphasizing the need for precise spatio-temporal coordination. Existing meth-

ods often prioritize motion cues from the dominant hand or jointly segment both hands' movements without explicitly accounting for different coordination types, resulting in insufficient segmentation for learning task constraints and modeling bimanual coordination.

To address these challenges, we propose a bottom-up keypoint-based hierarchical motion segmentation algorithm in Chapter 6 that create oversegmentation based on heuristics and subsequently merge them into semantically meaningful action primitives. Different from threshold-based approaches in the literature, our approach employs distance thresholds derived from object size to reduce the effort in threshold selection.

Furthermore, we adopt an object representation based on a set of densely tracked keypoints, accommodating both rigid and deformable objects while effectively capturing translational and rotational motion characteristics. We segment motions for each object pair in local frames anchored to one ojbect, creating an object-centric viewpoint alignment, elimitating embiguities encountered by approaches using a single global frame.

In summary, our proposed method offers a bottom-up, unsupervised, hierarchical approach to motion segmentation, leveraging keypoint-based object representation. This approach aims to achieve reliable motion segmentation with consistent granularity for complex bimanual tasks, facilitating the learning of task constraints and bimanual spatio-temporal coordination.

CHAPTER 3

Fundamentals

In this thesis, we construct an object-centric representation that leverages neural descriptors and dense visual correspondences to enable local pose estimation, keypoint extraction, and transfer across similar objects. This representation is a core component of the proposed KVIL framework, significantly enhancing intra-category generalization. The fundamentals of dense neural descriptors are introduced in Section 3.1.

Another critical component contributing to KVIL's generalization capabilities is the Principal Manifold Estimation (PME) algorithm (Meng and Eloyan, 2021). PME is used to uncover nonlinear geometric constraints from sets of 3D points by extracting a low-dimensional embedding – referred to as the principal manifold (e. g., curves and surfaces) – and predicting their reliable expansion trends. These capabilities are essential for representing and extrapolating constraints. We review the mathematical foundation of the PME algorithm in Section 3.2.

Finally, we introduce Via-point Movement Primitives (VMPs) in Section 3.3. These primitives learn the motions of individual keypoints and adapt the corresponding trajectories to novel via-points or goal points, which are identified in new scenes using the dense neural descriptors.

3.1. Dense Visual Descriptor

We investigate three categories of neural descriptors for constructing the object canonical space: 1) 2D neural descriptors learned via self-supervised training, 2) 3D neural descriptors obtained through similar methods, and 3) foundation–model-based neural descriptors.

Building on pioneering works such as Dense Object Net (DON) proposed in (Florence et al., 2018), Neural Descriptor Fields (NDF) proposed in (Simeonov et al., 2022a), and Neural Interaction Field and Template (NIFT) proposed in (Huang et al., 2023), we extract 2D and 3D neural descriptors from the first two categories. These approaches require the collection of data and model training for each or several object categories. The methodological foundations are detailed in Section 3.1.1 and Section 3.1.2, respectively.

In parallel, we utilize off-the-shelf models such as DINOv2 (Oquab et al., 2024) and Stable Diffusion (Tang et al., 2023) to extract foundation-model-based features. Pre-trained on large-scale datasets, these models facilitate direct feature extraction in an open-vocabulary setting. We discuss these foundation-model-based features in Section 3.1.3.

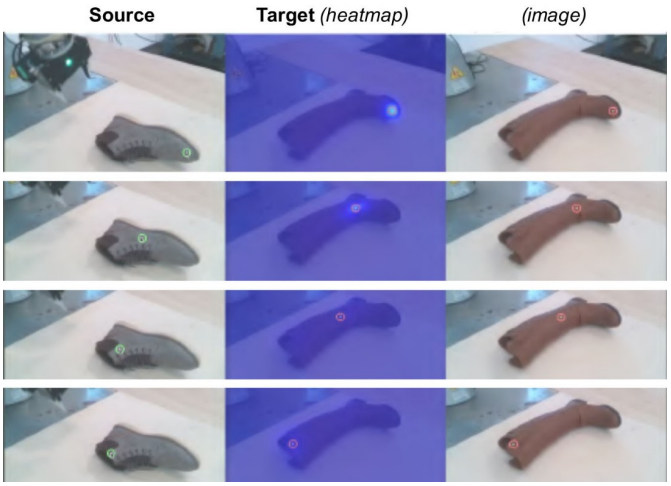
3.1.1. Self-supervised Training for 2D Features

Inspired by seminal works on dense visual correspondence (Choy et al., 2016; Schmidt et al., 2017), DON was introduced to learn dense visual descriptors for object categories, an approach first applied to robot manipulation tasks. DON learns a mapping function that maps an input RGB image $\mathbf{A}$ into a dense descriptor image $\mathbf{A}_D$,

$$\mathbf{A}_D = f_\theta^{\text{don}}(\mathbf{A}) \qquad \mathbf{A} \in \mathbb{R}^{W \times H \times C}, \mathbf{A}_D \in \mathbb{R}^{W \times H \times D}, \tag{3.1}$$

where W, H, C denote the width, height, and number of channels of the image, respectively, and D is the dimensionality of the descriptor space. Consequently, each pixel of the input image is represented by a D-dimensional descriptor $\mathbf{d} \in \mathbb{R}^D$.

A well-trained DON model maps similar local patches from images of objects within the same category to neighboring regions in descriptor space. In essence, a dense visual correspondence is established when the distance between two descriptors falls below a predefined threshold. When trained on



Note: Reprinted from Florence et al. (2020). © 2020 IEEE.

Figure 3.1.: Dense correspondences detected by DON. Given a selected pixel in the left image, DON finds the corresponding pixel in the right image that exhibits the highest cosine similarity in descriptor space. The middle column displays a heatmap of the cosine similarity between the selected pixel and all pixels in the right image.

intra-category image datasets, DON produces class-consistent descriptors. These descriptor similarities can be visualized as a heatmap showcasing dense correspondences between images (see Figure 3.1). Such representation enables downstream robotic tasks to utilize keypoint-based representations for object manipulation and skill transfer. For instance, given a keypoint on a reference image of a shoe as a grasping point, DON can be used to find the grasping point on similar shoes, allowing transferring picking up and arranging skills (Florence et al., 2018). Leveraging the dense properties of DON, Manuelli et al. (2020) incorporated DON in a model predictive control framework to control a set of dense keypoints following a reference trajectory.

In the original implementation, a convolutional neural network (CNN) serves as the mapping function f_θ^{don} to learn the dense visual descriptors. Specifically, various ResNet architectures (He et al., 2016) have been explored to serve as the backbone for DON, yielding precise dense correspondences between objects for manipulation tasks (Florence et al., 2018, 2020; Manuelli et al., 2020). Notably, this approach performs well for both rigid and deformable objects, even when significant shape variations are present. It can be extended to new object categories via retraining.

To train a DON model for a given object category, multiple views of posed RGB images of various object instances are collected. Florence et al. (2018) employed an RGB-D camera mounted on a robot arm to capture images from

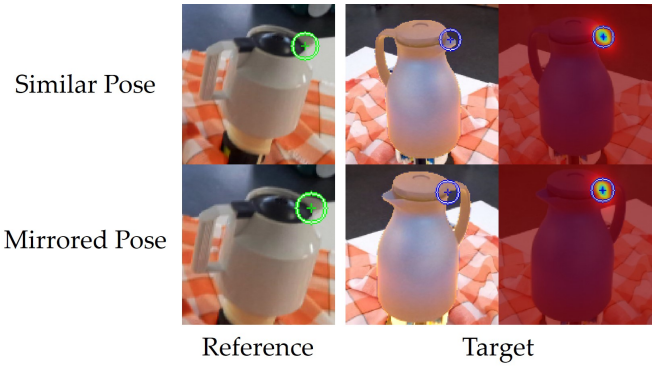


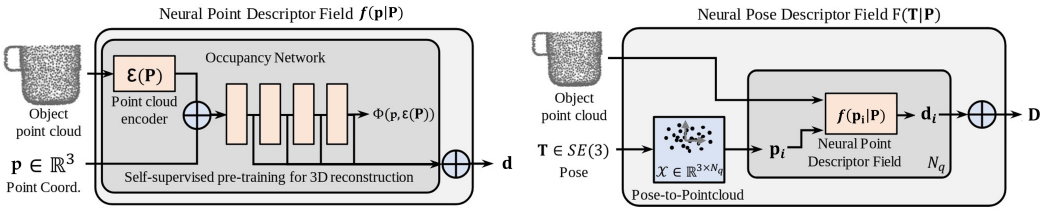
Figure 3.2.: The limitations of DON. DON detects precise correspondence of spout when reference and target objects have similar poses. However, when the target object is mirrored, DON fails to distinguish the handle from the spout.

diverse viewpoints, using robot forward kinematics to estimate camera extrinsic parameters for 3D reconstruction. In situations where wrist-mounted cameras are unavailable, Yen-Chen et al. (2022) proposed using an external camera (e. g., a smartphone) combined with Neural Radiance Field (NeRF, Mildenhall et al., 2021), for 3D reconstruction. However, NeRF-based methods typically require hours or days of training. In general, any state-of-the-art scene reconstruction technique is applicable. Inspired by Ichnowski et al. (2021), we modified Instant Neural Graphics Primitives (Instant-NGP, Müller et al., 2022) to perform rapid 3D reconstruction and depth inference, achieving scene reconstruction in under two minutes.

Despite their effectiveness, DON tends to over-emphasize the geometric aspect of the object on the semantic understanding, especially when part of the object is symmetrical. As shown in Figure 3.2, DON reliably detects the correspondence points on the spout of reference and target kettles if their poses are similar, which is also the case between the shoes in Figure 3.1. However, when the target kettle is mirrored, the detected correspondence points preserves similar spatial distribution in the image space but overlooks the semantic differences between the handle and the spout.

3.1.2. Self-supervised Training for 3D Features

To apply 2D image features in 3D object manipulation tasks, it is necessary to lift these features into 3D space using depth information. In scenarios where only a single viewpoint is available, as is common with humanoid robots, the detected keypoints are confined to the observable portion of the scene. This limitation can be problematic for tasks that require a comprehensive 3D understanding of



(a) Point descriptor (b) Pose descriptor

Note: Adapted from Simeonov et al. (2022a). © 2022a IEEE.

Figure 3.3.: The architecture of the Neural Descriptor Field (NDF). (a) A *point descriptor* represents the spatial features of a 3D point $\mathbf{p}$ relative to the object point cloud $\mathbf{P}$. (b) The descriptors of a set of 3D points collectively represents a pose in 3D space, named *pose descriptor*.

the object. To overcome this, recent approaches have exploited features directly extracted from 3D neural fields for applications such as grasping and object rearrangement, especially under conditions of partial occlusion.

Neural-fields-based approaches involve training neural networks to learn continuous representations of objects by predicting the physical and spatial properties of a 3D point relative to its local environment (Xie et al., 2022; Simeonov et al., 2022a; Huang et al., 2023). These learned models support various tasks, including object or scene reconstruction (Mescheder et al., 2019; Park et al., 2019) and object manipulation (Pfrommer et al., 2021; Karunratanakul et al., 2020). Their dense correspondence capabilities further enable the transfer of manipulation skills between similar objects.

Point descriptor: A *Neural Descriptor Field* (NDF) as described in Simeonov et al. (2022a) aims to learn a function f_θ^{NDF} that maps an arbitrary 3D point $\mathbf{p} \in \mathbb{R}^3$ to a feature vector $\mathbf{d} \in \mathbb{R}^D$, which encodes its spatial relationship with respect to a given object point cloud $\mathbf{P} \in \mathbb{R}^{3 \times N_p}$ with N_p points:

$$f_\theta^{\text{NDF}}(\mathbf{p}|\mathbf{P}) : \mathbb{R}^3 \times \mathbb{R}^{3 \times N_p} \rightarrow \mathbb{R}^D. \tag{3.2}$$

Unlike 2D descriptors, the 3D point $\mathbf{p}$ does not need to reside on the object surface or be visible from the camera. The resulting feature vector $\mathbf{d}$ serves as a point descriptor for the 3D point $\mathbf{p}$ relative to the object point cloud $\mathbf{P}$.

This descriptor is learned by training an SO(3)-Equivariant occupancy network (Mescheder et al., 2019; Deng et al., 2021) on a dataset of mean-centered object point clouds for category-level 3D reconstruction. The trained model predicts spatial occupancy from incomplete point clouds, and object shapes can subsequently be extracted using Multi-resolution IsoSurface Extraction

(MISE) (Mescheder et al., 2019). MISE incrementally constructs an octree to extract high-resolution meshes without exhaustively evaluating every point in a high-dimensional occupancy grid. Feature vectors, extracted from intermediate layers of the network's decoder, serve as point descriptors (see Figure 3.3a). The learned descriptor captures intrinsic geometric features of an object category – an emergent property of self-supervised training – ensuring that similar point clouds yield similar features in the descriptor fields. Furthermore, the SO(3)-Equivariant property guarantees consistent mapping under arbitrary object orientations.

Pose descriptor: The NDF framework can be extended to yield a pose descriptor by concatenating the point descriptors of a set of N_q fixed query points $\mathbf{P}_q \in \mathbb{R}^{3\times N_q}$ that represent the pose of a rigid object (see Figure 3.3b). These query points are typically sampled as a Basis Point Set (BPS, Prokudin et al., 2019). A rigid body pose $\mathbf{T}_r$ relative to an object point cloud $\mathbf{P}$ is then encoded via the following mapping:

$$\mathbf{D}_r = f_\theta^{\text{pose}}(\mathbf{T}_r|\mathbf{P}) = \bigoplus_{\mathbf{p}_i \in \mathbf{P}_q} f_\theta^{\text{NDF}}(\mathbf{T}_r\mathbf{p}_i|\mathbf{P}), \tag{3.3}$$

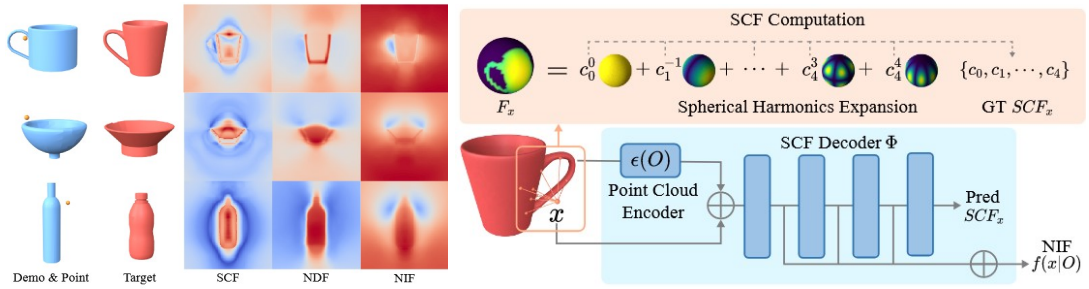
where $\bigoplus$ denotes the concatenation operator and $\mathbf{D} \in \mathbb{R}^{D\times N_q}$ is the resulting category-level pose descriptor.

Pose transfer: Given the point cloud $\mathbf{P}'$ of a new object instance of the same category f_θ^{NDF} is trained on, we can obtain a matching pose relative to $\mathbf{P}'$ resembling the spatial relationship between $\mathbf{T}_r$ and $\mathbf{P}$, by

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \| f_\theta^{\text{pose}}(\mathbf{T}|\mathbf{P}') - \mathbf{D}_r \|_1. \tag{3.4}$$

When the query points are sampled on robot gripper or humanoid hands, pose $\mathbf{T}^*$ corresponds to a target grasp pose, while when they are sampled on object, pose $\mathbf{T}^*$ then defines the target pose of that object relative to point cloud $\mathbf{P}'$.

As illustrated in Figure 3.4a, NDF descriptors sometimes struggle to distinguish specific regions of an object, such as the handle-side versus the non-handle-side of a mug or the top versus the bottom of a bottle. To address this limitation, Huang et al. (2023) introduced the Neural Interaction Field (NIF), which predicts the geometric features of a 3D point in the frequency domain via spherical harmonics (see Figure 3.4). The NIF is formulated as a normalized spherical function $f_{\mathbf{x}}^{\text{sp}}$ that aggregates the interception distances $d_{\mathbf{x}}$ of rays emitted from a



(a) Descriptor Fields (b) NIFT Architecture

Note: Reprinted from Huang et al. (2023). © 2023 IEEE.

Figure 3.4.: Comparison of neural descriptor field models and the architecture of NIFT. (a) shows the feature differences of a reference point to all the points around the target object for different neural descriptor models, visualized as a heatmap. (b) illustrates the NIFT architecture, which is similar to NDF but differs in the output of the prediction head, incorporating SCF features.

point in all directions by the object surface:

$$f_{\mathbf{x}}^{\mathrm{sp}}(\theta, \phi) = \frac{d_{min} + d_{avg}}{d_{\mathbf{x}}(\theta, \phi) + d_{avg}}, \tag{3.5}$$

where (θ, ϕ) denotes the spherical polar coordinates, and d_{min} and d_{avg} represent the minimum and average non-infinite intersection distances, respectively. This function can be decomposed via spherical harmonic expansion (Kazhdan et al., 2003):

$$f_{\mathbf{x}}^{\mathrm{sp}}(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_l^m Y_l^m(\theta, \phi), \tag{3.6}$$

where Y_l^m are the spherical harmonics at frequency l and a_l^m are the corresponding coefficients. Due to the rotation-invariant property of the L2-norm of the spherical harmonic function, the energy of each frequency component c_l is computed as:

$$c_l = \left\| \sum_{m=-l}^{l} a_l^m Y_l^m \right\|_2. \tag{3.7}$$

By restricting the frequency to n, the frequency information of a point $\mathbf{x}$ is summarized to its space coverage feature (SCF) defined as:

$$\mathsf{SCF}_x = \{c_0, c_1, ..., c_n\}. \tag{3.8}$$

See a visualization of SCF in Figure 3.4b. Within the NIF framework, the final layer of the NDF's occupancy network is modified to predict the SCF while preserving the remaining network architecture. Because the SCF encapsulates richer information, the learned descriptor becomes more distinctive and representative of category-level geometric relationships compared to standard NDF, as shown in Figure 3.4a.

While NIF provides more informative neural descriptors compared to NDF, it sacrifices the ability to reconstruct object shapes. This is because the shape reconstruction head in NDF is replaced with an SCF head in NIF, which does not support shape reconstruction. NIF performs well for grasp transfer when multiple views and a limited number of demonstrations (5-10) are available. However, its accuracy diminishes with partial views or when only a single demonstration is provided (Cai et al., 2024). To integrate the strengths of both approaches, we propose a novel implicit model that predicts multiple spatial features of a point relative to an object, including occupancy, signed distance, spherical harmonics, and the direction to the closest surface point. The objective is to construct a more informative descriptor space while preserving shape reconstruction capabilities. Our approach outperforms both NDF and NIF in tasks such as shape similarity measurement, pose transfer, and one-shot imitation learning of manipulation tasks, particularly under partial observation conditions (see Section 4.1.2).

Next, we explore a recent trend in extracting 2D neural descriptors from pre-trained large foundation models, which offer robust and general feature extraction capabilities in an open-vocabulary setting.

3.1.3. Features Extracted from Foundation Models

In addition to self-supervised training methods, we also explore the extraction of visual descriptors from off-the-shelf visual foundation models. These models, pre-trained on large-scale datasets, enable robust and general feature extraction in an open-vocabulary setting. Many such models exist with each focusing on different aspects of visual understanding.

The neural descriptors extracted from DINOv2 (Oquab et al., 2024) can identify semantic correspondences between objects, such as detecting the handle of a mug or the spout of a teapot. However, DINOv2 lacks detailed spatial information (Zhang et al., 2024b), which can result in ambiguities between the left and right sides of an object or between its front and back. As illustrated in Figure 3.5, the semantic descriptors effectively distinguish the semantic parts of a kettle –

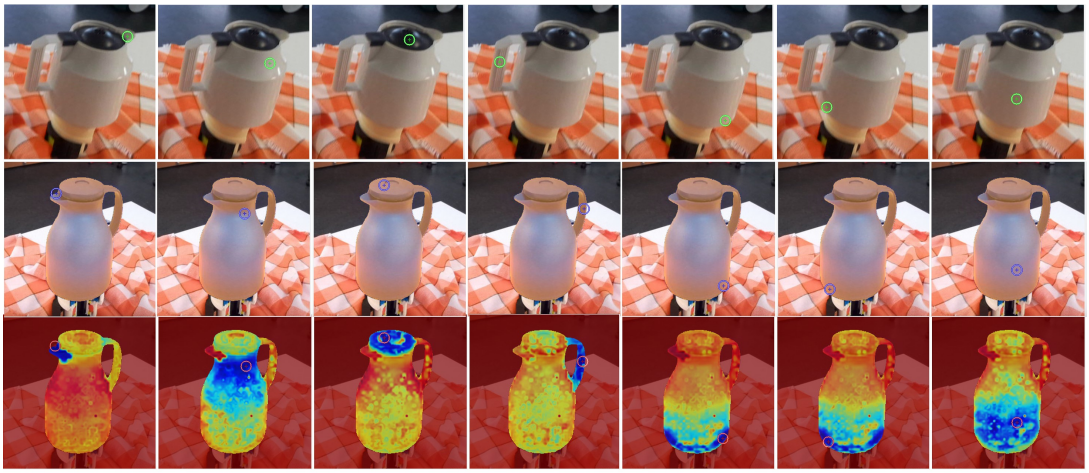


Figure 3.5.: Semantic descriptors of the DINv2 model are illustrated using kettle images. The top row shows a reference image where points are selected on different semantic parts of a kettle. The middle row displays the best matching correspondences for these points on a different kettle in a target image. The bottom row presents heatmaps showing the cosine similarity between the descriptor of the selected point in the reference image and the descriptors of all pixels in the target image. Blue indicates higher similarity, while red represents lower similarity.

including the handle, lid, neck, body, and bottom. Notably, even when the reference and target kettles exhibit nearly mirrored poses, the semantic descriptors can still correctly identify corresponding points. These descriptors tend to emphasize clear semantic boundaries between object parts (as indicated by the dark blue areas in Figure 3.5), which, while beneficial for semantic understanding, may obscure subtle spatial differences.

In contrast, emerging features from self-supervised Transformer and Diffusion models (DIFT, Tang et al., 2023) offer a smoother feature space that better captures spatial information. As shown in Figure 3.6, when the reference and target kettles have similar poses relative to the camera, the cosine similarity between DIFT features allows accurate correspondence matching. The DIFT feature space exhibits continuous and smooth variations, which are advantageous for distinguishing spatial contexts such as left versus right. However, it generally lack strong semantic cues, potentially leading to erroneous correspondence when object poses differ significantly. For instance, when the target kettle is mirrored (see Figure 3.7a), DIFT features may overly emphasize spatial context, resulting in mismatches in regions like the spout, handle, or sides of the bottom.

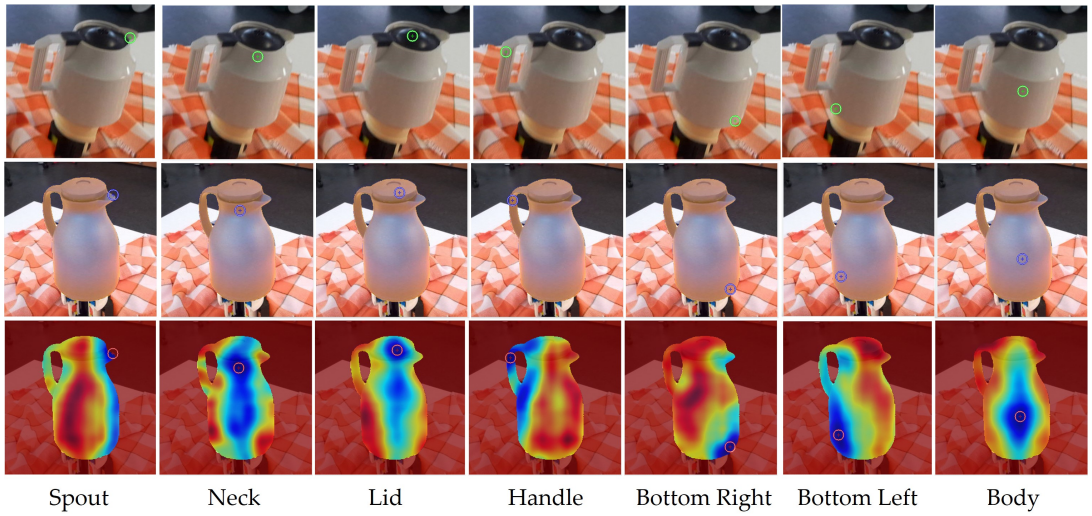
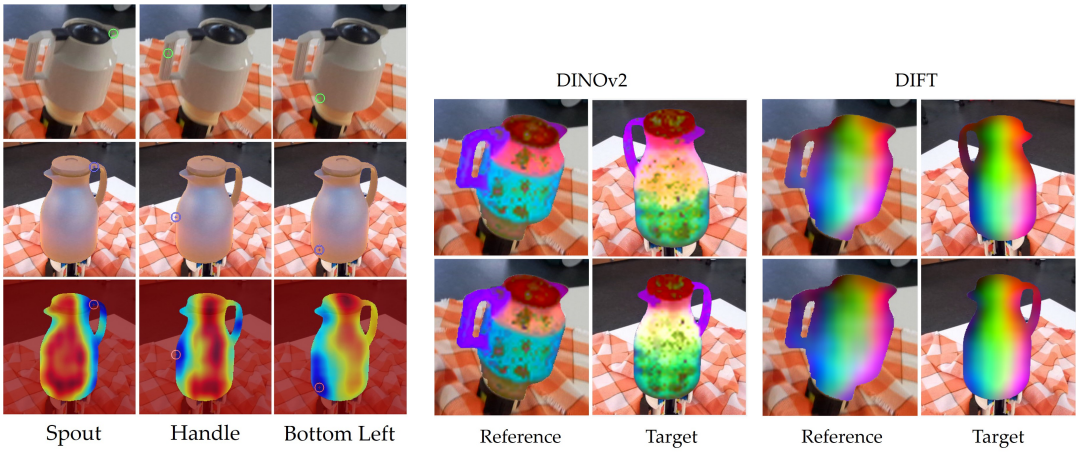


Figure 3.6.: Geometric descriptors of the DIFT model. The descriptions of point selection, correspondence detection and colored similarity map resembles that in Figure 3.5.

These differences are also evident in the low-dimensional visualizations shown in Figure 3.7b. DINOv2 features form distinct clusters corresponding to semantic parts of the kettle that remain consistent even under mirroring. In contrast, DIFT features appear more continuous and primarily capture spatial distributions (e. g., left-right, top-down), but they lack clear semantic separation, similarly to DON.

Recently, Ranzinger et al. (2024) introduced a novel foundation model, which distills the capabilities of several teacher models – including DINOv2, Stable Diffusion, and CLIP (Contrastive Language-Image Pre-training, Radford et al., 2021) – into a single framework. However, in many manipulation tasks evaluated in this thesis, these distilled models do not outperform their individual teacher models or their combinations. Consequently, we investigate a hybrid approach that combines semantic features (e. g., from DINOv2) with spatial features (e. g., from DIFT) to develop *geometry-aware semantic descriptors* (Zhang et al., 2024b). This integrated approach aims to enhance generalization in object manipulation tasks by leveraging the strengths of both semantic and spatial representations (see Section 4.1.1).



(a) DIFT features

(b) Colorized features

Figure 3.7.: A comparison of DINOv2 and DIFT features. (a) Correspondence detection using DIFT features between reference and target kettles, with the target kettle mirrored in pose. (b) Visualization of DINOv2 and DIFT features from both reference and target images. The features are projected into a 3D space using PCA, with the normalized dimensions of the projection mapped to RGB channels for visual representation.

3.2. Principal Manifold Estimation (PME)

Within the KVIL framework, our objective extends beyond extracting a set of keypoints to also learning the geometric constraints that these keypoints must satisfy in order to represent a task. In this context, geometric constraints are extracted along with the keypoints (see Figure 1.2b). In three-dimensional space, such a constraint can be modeled as a low-dimensional manifold corresponding to a point, line, plane, curve, or surface. These low-dimensional embeddings are typically estimated using dimensionality reduction or manifold learning techniques, as reviewed in Section 2.1.3.

Among the methods, the Principal Manifold Estimation (PME) algorithm (Meng and Eloyan, 2021) is notable for its ability to extract both linear and nonlinear manifolds from sparse data and to predict their expansion trends reliably – an advantage that is particularly useful for extrapolation. Despite this strength, PME has not been applied in robotic manipulation tasks.

In this thesis, we leverage the PME algorithm to uncover geometric constraints as low-dimensional embeddings from sets of 3D keypoints observed in human demonstrations. In PME, the principal manifold is defined as a minimum of the functional with a regularity penalty term derived on a Sobolev space. Specifically,

the PME algorithm minimizes the loss function

$$\mathcal{L}(f_r, \pi_d) = \mathbb{E}\,\|\mathbf{p} - f_r(\pi_d(\mathbf{p}))\|^2 + \lambda\|\kappa_f\|^2, \tag{3.9}$$

where $\pi_d : \mathbb{R}^D \to \mathbb{R}^d$ is the *projection index* that maps a D-dimensional vector $\mathbf{p}$ onto a d-dimensional principal manifold (with $d < D$), $f_r : \mathbb{R}^d \to \mathbb{R}^D$ is the *reconstruction function*, $\|\kappa_f\|^2$ represents the high-dimensional generalization of the total squared curvature of the principal manifold. The parameter and $\lambda \in [0, \infty)$ controls the model complexity. In this formulation, the first term quantifies the reconstruction error, while the second term regularizes the model to mitigate overfitting.

It is noteworthy that as $\lambda \to \infty$, PME converges to linear PCA. The linearity and the dimensionality d of the principal manifold determine the type of subspace. For example, in 3-dimensional world ($D = 3$), a nonlinear principal manifold with $d = 1$ corresponds to a principal curve (i. e., a curve constraint), whereas with $\lambda \to \infty$ and $d = 1, 2$, the principal manifold reduces to a principal line or plane, respectively (i. e., linear constraints). These linear principal manifolds essentially reduce to the principal components obtained via PCA. Given that PME is an iterative algorithm, its computation cost is higher than that of PCA; therefore, for linear constraints, PCA may serve as a more efficient alternative. Further details on constraint extraction are provided in Section 4.2.

3.3. Via-point Movement Primitive (VMP)

In addition to extracting keypoint constraints, we aim to learn their associated motions from human demonstration videos. For example, consider the motion of a kettle spout represented in a local frame attached to the opening of a cup. In imitation learning, movement primitives are widely used to represent and reproduce motion patterns (Osa et al., 2018). Here, we adopt Via-point Movement Primitives (VMPs) (Zhou et al., 2019) because they enable learning from multiple demonstrations of keypoint trajectories and robustly adapting these trajectories to novel via-points, thereby supporting effective extrapolation.

A VMP models a trajectory as the sum of a linear elementary component h_{vmp} and a nonlinear shape modulation f_{vmp}:

$$y(x) = h_{\text{vmp}}(x) + f_{\text{vmp}}(x) = g + x(y_0 - g) + \boldsymbol{\psi}(x)^\mathsf{T}\boldsymbol{w},$$

where x is the canonical variable decreasing linearly from 1 to 0, y is the position at canonical variable x, y_0 is the initial position, g is the target position, and $\boldsymbol{w}$ is the weight parameters. The shape modulation term is defined as a linear regression model based on N_k squared exponential (SE) kernels:

$$\psi_i(x) = \exp(-h_i(x - c_i)^2), i \in [1, N_k], \tag{3.10}$$

where h_i, c_i are pre-defined constants representing the width and center of the kernels. Similarly to probabilistic movement primitives (ProMP) (Paraschos et al., 2018), VMPs assume that the weight parameters $\boldsymbol{w}$ follow a Gaussian distribution $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu_w}, \boldsymbol{\Sigma_w})$, and can thus be learned via maximum likelihood estimation (MLE). VMPs provide enhanced extrapolation capability compared to ProMP, as they handle via-points (including start and target positions) adaptation to points that lie out of the demonstrated distributions.

In this thesis, VMPs are employed to learn the demonstrated motion styles of individual keypoints and adapt their trajectories to novel via-points identified using dense neural descriptors. In contrast to keypoint-based visual imitation learning approaches that rely on reinforcement learning control policies (Sieb et al., 2019) or visual servoing techniques (Jin and Jagersand, 2022), VMP-based control policies offer flexible temporal scaling and reliable via-point adaptation, while being able to effectively learn from a few demonstrations.

In the next chapter, we present the proposed KVIL approach, which integrates dense neural descriptors, Principal Manifold Estimation, and Via-point Movement Primitives to learn keypoint-based task representations from video demonstrations and to reproduce the learned manipulation skills on humanoid robots in various scenarios.

CHAPTER 4

Learning Keypoint-based Task Representation

This chapter addresses the fundamental challenges outlined in the first research question: *learning subsymbolic keypoint-based task representations* from sparse human video demonstrations of manipulation tasks. These challenges include: 1) developing neural descriptor-based, generalizable object representations; 2) modeling keypoint-based task constraints; 3) effectively extracting these task constraints; and 4) formulating keypoint-based control policies for task reproduction.

To address the first challenge, we develop robust object representations using neural descriptors. While existing methods show promise (see Section 2.1.1), they often struggle with generalization in everyday manipulation tasks due to inconsistencies in geometric and semantic meanings within the descriptor space. In Section 4.1, we propose solutions that overcome these limitations by extracting robust keypoint-based object representations from human demonstration videos.

Building on these object representations, we tackle the second core challenge in Section 4.2 by modeling keypoint-based geometric constraints on principal manifolds as a fundamental component of generalizable task representation. We define geometric constraints as subspaces in our 3D world, encompassing points, lines, planes, curves, surfaces, poses, and their combinations. Critically, these constraints are expressed relative to local frames anchored to specific object parts, enabling object-centric task representation and hierarchical scene decomposition for enhanced generalization capabilities.

For the third core challenge, we employ statistical methods – specifically Principal Component Analysis (PCA, Pearson, 1901) and Principal Manifold Estimation algorithms (PME, Meng and Eloyan, 2021) – to identify and extract invariant features (i. e., keypoints and constraints) across multiple human demonstrations (see Section 4.2). Our methodology begins with *densely* sampling candidate points on objects and leveraging dense neural descriptors to track their trajectories throughout demonstration videos (see Section 4.1.4). Subsequently, we jointly extract a set of *sparse keypoints* from these dense candidates, their associated *object-centric local frames*, and *geometric constraints* that collectively represent the task, as illustrated in Figure 1.2b.

The representation of tasks through keypoints and their constraints enables the implementation of movement primitives for task execution, addressing our fourth core challenge. We encode keypoint motions relative to their corresponding local frames as Via-point Movement Primitives (VMPs, Zhou et al., 2019), which offer flexibility in temporal scaling and trajectory adaptation while preserving the style of demonstrated motions. These learned keypoint motions can then be executed on a robot using our novel keypoint-based admittance controller (see Section 4.3).

We validate our approach by learning various daily tasks from video demonstrations and reproducing them with a humanoid robot (see Section 4.4). The results demonstrate that KVIL efficiently extracts generalizable manipulation skills, handles viewpoint mismatches, and effectively manages large pose and shape variations of categorical objects in cluttered scenes.

The contributions of this chapter are summarized in Section 1.2.1. Parts of the content presented in this chapter were published by Gao et al. (2023) and Cai et al. (2024). As this chapter focuses specifically on unimanual tasks, we refer to our approach as Uni-KVIL, which will be extended to bimanual manipulation tasks in Chapter 5 and a series of sub-tasks in Chapter 6.

4.1. Generalizable Object Representation

Object representation plays an essential role in establishing generalizable task representations for robotic manipulation (see Sections 2.1.1 and 2.1.5). Our primary objective in this section is to establish a generalizable object representation by leveraging 2D and 3D neural descriptors in Sections 4.1.1 and 4.1.2, respectively. We then construct a canonical space per object category to facilitate knowledge transfer across different object instances (see Section 4.1.3). Finally,

we introduce a perception pipeline utilizing the neural descriptors, object canonical space, and other state-of-the-art computer vision techniques to detect and track dense points on objects and human hands, effectively handling challenging occlusions (see Section 4.1.4). This preprocessing pipeline provides high-quality point-based data for subsequent modeling of keypoint-based constraints and their extraction algorithms in Section 4.2.

4.1.1. Neural Descriptors Derived from RGB Images

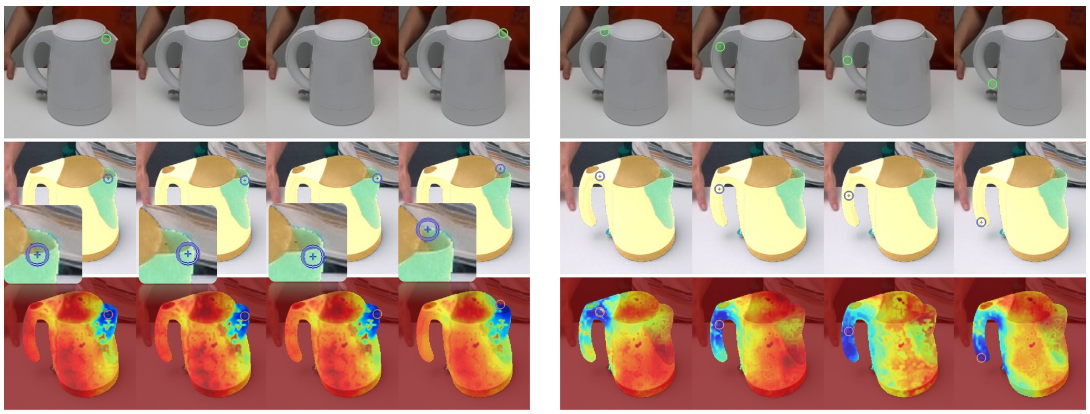
As detailed in Section 3.1, neural descriptors derived from RGB images establish dense correspondences between objects and their constituent parts, which is fundamental for achieving viewpoint invariance and intra-category generalizations. Despite promising advances in the literature, applying these techniques to daily manipulation tasks reveals significant challenges.

A typical limitation of geometric descriptors – such as DON and DIFT features – is their inability to capture semantic information of objects. This limitation becomes more pronounced when reference and target objects have mirrored poses, as illustrated in Figure 3.2 and Figure 3.7. Conversely, pure semantic features – such as DINO – lack geometric information; they offer clear boundaries between object semantic parts but cannot produce smooth feature spaces or distinguish spatial differences between semantically similar points, as illustrated in Figure 3.5.

To address these limitations, we follow Zhang et al. (2024b) in combining the geometric features of DIFT (Tang et al., 2023) with the semantic features of DINOv2 (Oquab et al., 2024) to create a more informative *geometry-aware semantic descriptor space*. Specifically, given an input image $\mathbf{A}$, the combined feature image $\mathbf{A}_D$ are obtained by concatenating the DINOv2 and DIFT descriptors:

$$\mathbf{A}_D = f^{\text{DINO}}(\mathbf{A}) \bigoplus f^{\text{DIFT}}(\mathbf{A}) \qquad \mathbf{A} \in \mathbb{R}^{W \times H \times 3}, \mathbf{A}_D \in \mathbb{R}^{W \times H \times D}, \tag{4.1}$$

where $D = D^{\text{DINO}} + D^{\text{DIFT}}$ is the dimensionality of the resulting descriptor space as a sum of the dimensionality of the DINOv2 and DIFT feature spaces. Note that, for simplicity, we refer these features derived from 2D RGB images "2D descriptors" and those derived from 3D data "3D descriptors", where "2D" and "3D" reflect the data modality instead of the dimensionality of the feature space. For simplicity, we denote the combined model f_x^{DD}, which maps each pixel $\mathbf{x}$ in the image to the combined descriptor space. Here, the superscript $^{\text{DD}}$ stands for the combination of **D**INOv2 and **D**IFT: The descriptor for a pixel using the



Spout Handle

Figure 4.1.: Geometry-aware semantic descriptors obtained by concatenating the DINOv2 and DIFT descriptors. Correspondence detection in fine-grained area of spout (left) and handle (right) parts of the kettle are shown. The descriptions of point selection, correspondence detection and colored similarity map resembles that in Figure 3.5.

combined model is

$$\mathbf{d} = f_x^{\text{DD}}(\mathbf{x}) = f_x^{\text{DINO}}(\mathbf{x}) \bigoplus f_x^{\text{DIFT}}(\mathbf{x}) \qquad \mathbf{x} \in \mathbb{R}^3, \mathbf{d} \in \mathbb{R}^D, \tag{4.2}$$

where $\mathbf{d}$ denotes the combined descriptor for pixel $\mathbf{x}$ and f_x denotes the corresponding descriptor model applied a specific pixel.

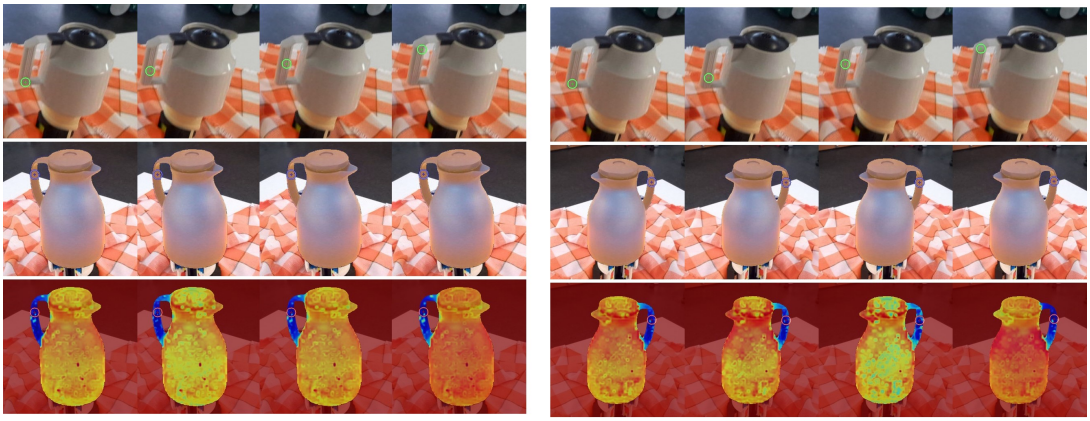
To detect correspondences between reference and target objects, we utilize the cosine similarity between two descriptors $\mathbf{d}^{\text{ref}}$ and $\mathbf{d}^{\text{tar}}$:

$$\mathsf{SIM}(\mathbf{d}_{\text{ref}}, \mathbf{d}_{\text{tar}}) = \frac{\mathbf{d}_{\text{ref}} \cdot \mathbf{d}_{\text{tar}}}{\|\mathbf{d}_{\text{ref}}\| \|\mathbf{d}_{\text{tar}}\|}. \tag{4.3}$$

The *Best Matching* (BM) point in the target image $\mathbf{A}_t$ is identified by finding the pixel with the highest cosine similarity to the reference descriptor:

$$\mathbf{x}_{\text{t}}^* = \mathsf{BM}(\mathbf{x}_{\text{r}}) = \arg\max_{\mathbf{x} \in \mathbf{A}_{\text{t}}} \mathsf{SIM}(f_x^{\text{DD}}(\mathbf{x}_{\text{r}}), f_x^{\text{DD}}(\mathbf{x})), \tag{4.4}$$

where $\mathbf{x}_r$ is the reference pixel, and $\mathbf{d}_r = f_x^{\text{DD}}(\mathbf{x}_{\text{r}})$ is the reference descriptor. To further constrain the detected correspondence within the target object region, we employ the GroundingDINO (Liu et al., 2024) object detection algorithm to provide the object bounding box, which is subsequently used by the Segment Anything Model (SAM, Kirillov et al., 2023) to generate object binary masks. The cosine similarity is then computed between the descriptors of the reference



Similar Pose Mirrored Pose

Figure 4.2.: The DINOv2 features are not able to distinguish different points within the handle of the kettle.

point and those within the target object's binary mask. Given the binary mask of the target object $\mathbf{M}_t$, the best matching point is determined by:

$$\mathbf{x}_t^* = \mathsf{BM}(\mathbf{x}_r, \mathbf{M}_t) = \arg\max_{\mathbf{x} \in \mathbf{M}_t} \mathsf{SIM}(f_x^{\mathrm{DD}}(\mathbf{x}_r), f_x^{\mathrm{DD}}(\mathbf{x})). \tag{4.5}$$

As demonstrated in Figure 4.1, given a selected pixel in a reference image of a kettle, the combined features successfully detect correspondences in fine-grained areas such as the spout and handle of the target kettle – regions that are challenging for individual features. In contrast, DINOv2 features struggle to distinguish different points within the handle in cases with similar and mirrored poses (see Figure 4.2), while DIFT features fail to correspond to semantic parts as shown in Figure 3.7.

Despite these improvements, the combined features still exhibit limitations including artifacts, uneven smoothness, viewpoint variance, and imbalanced contributions of DINOv2 and DIFT features in the descriptor space. For instance, as shown in Figure 4.3, when reference and target objects exhibit mirrored poses, the combined features outperform DIFT in locating handles but fail to smoothly distinguish between points on the handle due to the dominance of DINOv2 features. Although incorporating DIFT features enhances geometry-awareness in cases with similar poses (see Figure 4.1), at the bottom of the kettle, the combined features are dominated by DIFT features, resulting in difficulty distinguishing left from right. Furthermore, artifacts and missing smoothness in the combined descriptor space are evident in the kettle's body, where slight changes in the reference point lead to significant shifts in the detected correspondence point.

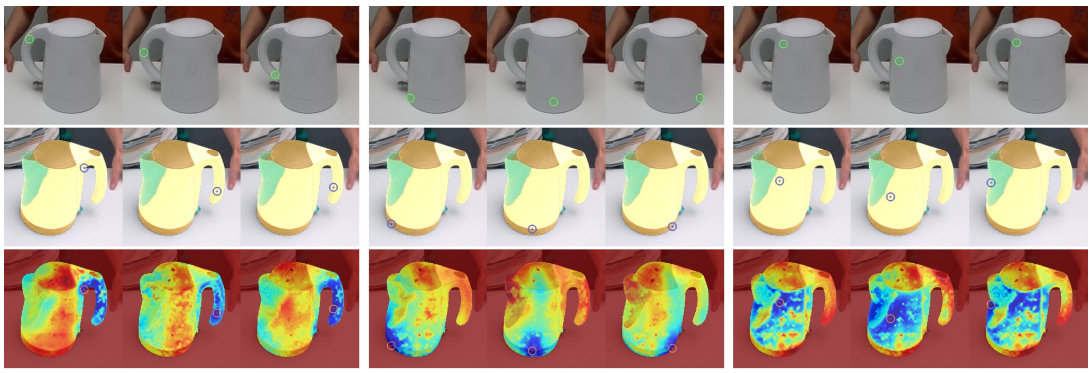


Figure 4.3.: Imprecise or incorrect correspondence of the combined DINOv2 and DIFT model for kettle's spout, handle and bottom part, when reference and target objects are mirrored.

These artifacts and discontinuities can also be observed in cases with similar poses.

Dense Correspondence Detection

The capability to detect potential keypoints at arbitrary locations on objects is essential for developing task representations at the finest granularity. This requires a *viewpoint-invariant, smooth, and geometry-aware semantic descriptor space* that can distinguish both semantic parts and fine-grained geometric details of objects, such as the top, middle, and bottom parts of a kettle's handle as shown in Figure 4.1. We present a four-step correspondence detection algorithm, involving: 1) viewpoint augmentation, 2) similarity-based retrieval, 3) sparse matching, and 4) dense warping. The goal of this algorithm is to reliably detect corresponding pixels on categorical objects in target images, regardless of viewpoint differences between reference and target images.

Viewpoint augmentation: To address the viewpoint variance problem, Zhang et al. (2024b) proposed applying viewpoint augmentation to images – including horizontal flips, double flips, and rotations of $+90°$, $180°$, and $-90°$ – to improve pose estimation accuracy. Similarly, Ausserlechner et al. (2024) employed multiple views of target objects to find target images with the best matching pose, followed by correspondence-based object pose estimation. However, augmenting target images at testing time requires computing features for all multi-view images, which is computationally expensive. We propose a pipeline with *offline augmentation of reference images* and an *online exhaustive matching* on a single target image. To achieve better performance for objects with arbitrary poses, we

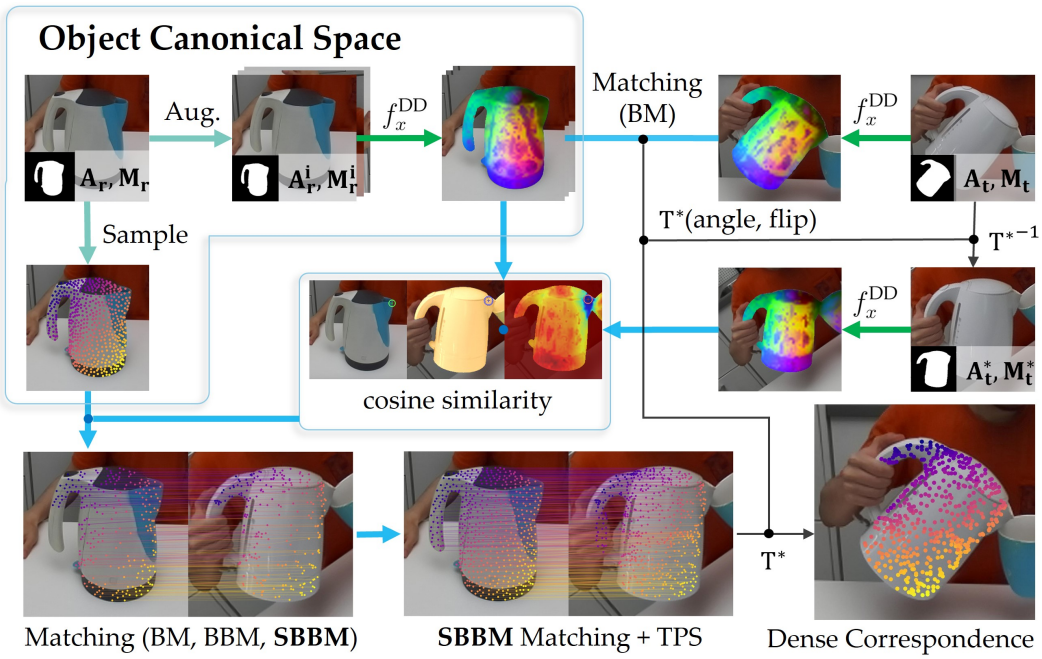


Figure 4.4.: The dense correspondence detection pipeline. Notations: $\mathbf{A}_\mathrm{r}, \mathbf{M}_\mathrm{r}$ are the reference images, $\mathbf{A}_\mathrm{r}^i, \mathbf{M}_\mathrm{r}^i$ are the augmentation images, $\mathbf{A}_\mathrm{t}, \mathbf{M}_\mathrm{t}$ and $\mathbf{A}_\mathrm{t}^*, \mathbf{M}_\mathrm{t}^*$ are the target and corrected target images. Connections show the estimation of descriptors (⟶), similarity-based correspondence detection (⟶), and image transformation (⟶). The color gradient of sampled reference points is computed based on the distance to the top-left corner of the image. Point colors of dense correspondence points are based on the identity matching to the reference point. Colors of the descriptor images are determined as in Figure 3.7.

extend the augmentation views by rotating reference images in $18°$ increments 20 times, covering the full $360°$ rotation. These images are subsequently flipped horizontally to account for mirrored cases. Augmentation details are presented in Appendix A.3.

Similarity-based retrieval: We propose a novel retrieval method based on image feature similarities and the number of matching pixels. The objective is to identify the best matching image from the augmented set of reference images for a given target image. Once identified, we transform the target image to align with the original reference image using the transformation parameters of the best matching augmented reference image.

As shown in Figure 4.4, we compute the combined descriptors for each augmented image within the masked area of the object. Given a test image of a different kettle on the top right, we obtain its descriptor image and apply a matching algorithm to retrieve the rotation and flipping augmentation parame-

ters that result in the most similar reference image. As an alternative to the Best Matching (BM) technique in Eq. (4.4) for correspondence detection, we improve cyclic consistency by using the *Best-Buddies Matching* (BBM) algorithm (Dekel et al., 2015). The BBM algorithm identifies pairs of points on the reference and target object's masked areas ($\mathbf{M}_\mathrm{r}, \mathbf{M}_\mathrm{t}$) where each point is the best match of the other:

$$\begin{aligned}\mathcal{X}_{\mathrm{BBM}} = \{(\mathbf{x}_\mathrm{r}, \mathbf{x}_\mathrm{t}) \,|\, \mathbf{x}_\mathrm{t} = \mathsf{BM}(\mathbf{x}_\mathrm{r}, \mathbf{M}_\mathrm{t}), \mathbf{x}_\mathrm{r} = \mathsf{BM}(\mathbf{x}_\mathrm{t}, \mathbf{M}_\mathrm{r}) \\ \forall \mathbf{x}_\mathrm{r} \in \mathbf{M}_\mathrm{r}, \mathbf{x}_\mathrm{t} \in \mathbf{M}_\mathrm{t}\}.\end{aligned} \tag{4.6}$$

For each augmented image ($\mathbf{A}^i$, $\mathbf{M}^i$), we compute the probability that it is a best-buddies matching of the target image in terms of the average similarity score and the number of best buddies corresponding to the detected set of points $\mathcal{X}^i_{\mathrm{BBM}}$. Specifically,

$$S^i = \frac{1}{|\mathcal{X}^i_{\mathrm{BBM}}|} \sum_{(\mathbf{x}^i_r, \mathbf{x}_t) \in \mathcal{X}^i_{\mathrm{BBM}}} \mathsf{SIM}\left(f_x^{\mathrm{DD}}(\mathbf{x}^i_r), f_x^{\mathrm{DD}}(\mathbf{x}_t)\right), \mathbf{x}^i_r \in \mathbf{M}^i \tag{4.7}$$

$$\mathsf{Pr}^i_{\mathsf{SIM}} = \frac{S^i}{\sum_{j=1}^{N_a} S^j} \tag{4.8}$$

$$\mathsf{Pr}^i_{\mathsf{num}} = \frac{|\mathcal{X}^i_{\mathrm{BBM}}|}{\sum_{j=1}^{N_a} |\mathcal{X}^j_{\mathrm{BBM}}|} \tag{4.9}$$

where S^i is the similarity score between the i-th reference image and the target image, $\mathsf{Pr}^i_{\mathsf{SIM}}$ and $\mathsf{Pr}^i_{\mathsf{num}}$ are the normalized probabilities of the similarity scores and the detected number of best buddies across all N_a augmented images. The best match is retrieved as

$$i^* = \underset{i \in [1, N_a] \cap \mathbb{Z}}{\arg\max}\, \mathsf{Pr}^i_{\mathsf{SIM}} \cdot \mathsf{Pr}^i_{\mathsf{num}}. \tag{4.10}$$

The corresponding augmentation transformation operator T^* is then used to inversely transform the target image to obtain the corrected viewpoint of the target image that best matches the original reference image, denoted $\mathbf{A}^*_\mathrm{t}$.

Sparse matching: We then propose a *Soft-Best-Buddies matching* (SBBM) algorithm to obtain sparse pixel correspondences between reference and transformed target images.

Subsequently, we employ the cosine similarity between the descriptors of the reference image $\mathbf{A}_\mathrm{r}$ and the corrected target image $\mathbf{A}^*_\mathrm{t}$ to find dense correspondence points. Given a set of points $\mathcal{P}^s_r$ on the reference object as shown in Figure 4.4, correspondence points on the target image can be estimated using BM or BBM

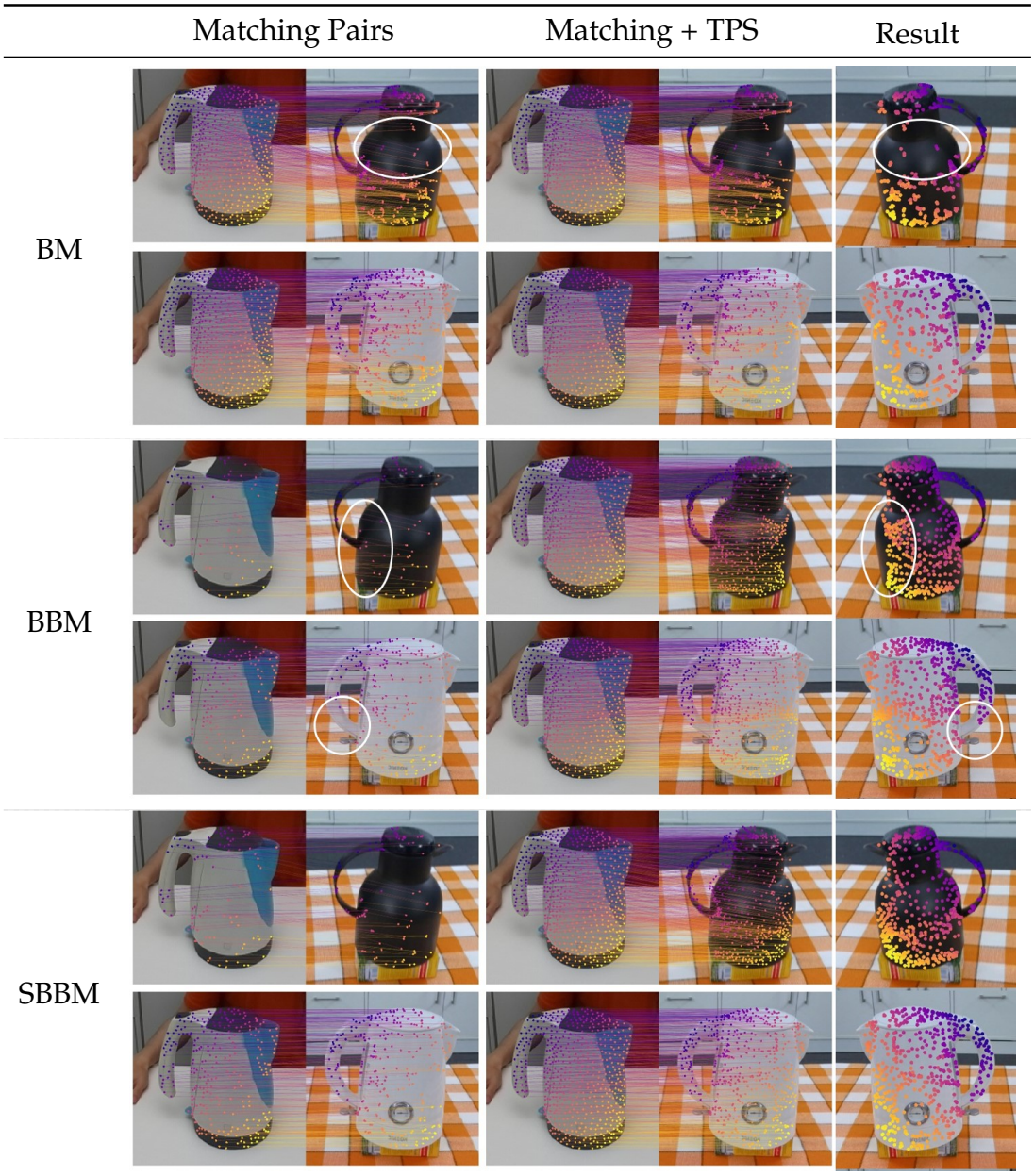


Figure 4.5.: Comparison of three different correspondence detection algorithms, namely Best Matching (BM), Best-Buddies Matching (BBM) and Soft Best-Buddies Matching (SBBM). The first image column depicts the matching point pairs of each algorithm, with the reference image on the left and the transformed target image on the right; The second image column shows the results of the combination of correspondence matching and TPS. Note that we use a target image (right most column) with a mirrored pose compared to the reference image to show that we can also handle cases with viewpoint mismatch; but the matching and TPS steps are applied on the transformed target image $\mathbf{A}_t^*$. The last column shows the resulting correspondence points on the original target image $\mathbf{A}_t$.

algorithms. The matching pairs of both algorithms for different target kettles are shown in Figure 4.5. However, due to artifacts and unsmoothness in the descriptor space, BM tends to find correspondence points clustered in small areas, while both BM and BBM fail to find enough correct correspondences in the areas marked with white circles.

To address this limitation, we propose the *Soft-Best-Buddies matching* (SBBM) algorithm that extends BBM by relaxing the strict requirement of being the absolute best match. Instead, it considers points whose cyclic best buddies are within a distance threshold ξ_{pixel} in image space:

$$\begin{aligned}\mathcal{X}_{\text{SBBM}} = \{(\mathbf{x}_\text{r}, \mathbf{x}_\text{t}) \mid \|\mathbf{x}_\text{r} - \mathsf{BM}(\mathbf{x}_\text{t}, \mathbf{M}_\text{r})\| < \xi_{\text{pixel}}, \\ \mathbf{x}_\text{t} = \mathsf{BM}(\mathbf{x}_\text{r}, \mathbf{M}_\text{t}) \quad \forall \mathbf{x}_\text{r} \in \mathbf{M}_\text{r}\}.\end{aligned} \tag{4.11}$$

Both BBM and SBBM improve cyclic consistency by enforcing a bidirectional matching constraint, which helps eliminate spurious matches and enhances the reliability of correspondences between points in the source and target sets. Though SBBM generally increases the number of sparse matching pairs, as shown in Figure 4.5, its result is not directly applicable for dense correspondence.

Dense warping: To address this problem, we propose a novel *dense correspondence detection algorithm* by integrating SBBM with a smoothing deformation field leveraging Thin-plate-spline (TPS, Duchon, 1977).

The TPS algorithm is a non-rigid deformation technique that warps the reference image to the target image by minimizing bending energy to ensure the smoothest possible deformation. The deformation field is guided by control points extracted from both the reference and target images, which are then used to compute the transformation warping the reference image to the target image. In our case, the control points are the set of matching pairs, such as $\mathcal{X}_{\text{BBM}}$ and $\mathcal{X}_{\text{SBBM}}$.

Given a randomly sampled pixel $\mathbf{x}'_r = (u, v)$ on the reference object, it can be lifted to a 3D point $\mathbf{p}_r$ via triangular geometry of the camera and the depth image $\mathbf{D}$,

$$\mathbf{p}_r = \mathsf{Lift}(\mathbf{x}'_r, \mathbf{D}(\mathbf{x}'_r)) = \mathbf{D}(\mathbf{x}'_r) \cdot \mathbf{K}^{-1} \Big(u, v, 1\Big)^\mathsf{T}, \quad \mathbf{p}_r \in \mathbb{R}^3 \tag{4.12}$$

where $\mathbf{K}$ is the camera intrinsic parameter. Similarly, the set of matching pixel pairs, such as the soft best-buddies $\mathcal{X}_{\text{SBBM}}$, are lifted to 3D by:

$$\mathcal{P}_{\text{src}} = \{\mathsf{Lift}(\mathbf{x}_r, \mathbf{D}(\mathbf{x}_r)) \mid (\mathbf{x}_r, \mathbf{x}_t) \in \mathcal{X}_{\text{SBBM}}\}, \tag{4.13}$$

$$\mathcal{P}_{\text{tgt}} = \{\mathsf{Lift}(\mathbf{x}_t, \mathbf{D}(\mathbf{x}_t)) \mid (\mathbf{x}_r, \mathbf{x}_t) \in \mathcal{X}_{\text{SBBM}}\}, \tag{4.14}$$

where $\mathcal{P}_{\text{src}}$ and $\mathcal{P}_{\text{tgt}}$ serve as source and target control points for the TPS algorithm. We then estimate the correspondence $\mathbf{p}_t$ on the target images by:

$$\mathbf{p}_t = \mathsf{TPS}(\mathbf{p}_r \mid \mathcal{P}_{\text{src}}, \mathcal{P}_{\text{tgt}}) \tag{4.15}$$

where TPS is the smooth deformation field conditioned on the control points $\mathcal{P}_{\text{src}}$ and $\mathcal{P}_{\text{tgt}}$, which can be replaced by matching pairs of BM or BBM.

As an application, assume we have a set of sampled pixels $\mathcal{X}_r$ evenly distributed on the reference object as shown in Figure 4.4. We lift them to 3D and apply the TPS operator using the same set of control points to obtain the dense correspondences $\mathcal{P}_t$ on the target images:

$$\mathcal{P}_r = \{\mathsf{Lift}(\mathbf{x}_r, \mathbf{D}(\mathbf{x}_r)) \mid \mathbf{x}_r \in \mathcal{X}_r\}, \tag{4.16}$$

$$\mathcal{P}_t = \left\{\mathsf{TPS}(\mathbf{p}_r \mid \mathcal{P}_{\text{src}}, \mathcal{P}_{\text{tgt}}) \mid \mathbf{p}_r \in \mathcal{P}_r\right\} \tag{4.17}$$

We denote the set of pixels obtained by projecting the 3D correspondence points $\mathcal{P}_t$ onto the target image as $\mathcal{X}_t$, which is depicted in the bottom-right corner of Figure 4.4.

In practice, the quality of the matching pairs significantly affects the deformation field of TPS. As shown in the second column of Figure 4.5, the TPS deformation field is applied to the reference image to warp the sampled set of dense reference points to the target image using different matching algorithms. Since all sampled points in the reference image are assigned a correspondence in the BM case (first row in Figure 4.5), making them the control points, applying TPS has little effect on the control points themselves. Therefore, clusters and holes remain. Comparing the results of BBM and SBBM, we clearly see that the additional control points provided by SBBM enhance the smoothness of the TPS field, yielding better dense correspondence, especially in the areas marked with white circles.

As shown in Figure 4.4, the dense correspondence results are then projected back to the original target image using the best matching transformation T^*. Similar results are depicted in the last column of Figure 4.5.

Summary: We proposed a novel dense correspondence algorithm between pixels in reference and target images. It consists of 1) *viewpoint augmentation* of reference image, 2) *retrieval* of best matching reference image and *correction* of target image based on feature similarity and the number of matching pixels, 3) *SBBM* and 4) *TPS-based dense warping*.

It is important to note that augmented reference images, their descriptors, and sampled reference points serve as important building blocks of the object canonical space, which is essential in building object-centric task representations, as will be detailed in Section 4.1.3. The four-step correspondence detection pipeline allows robust dense correspondence detection on categorical objects by leveraging emerging features in visual foundation models.

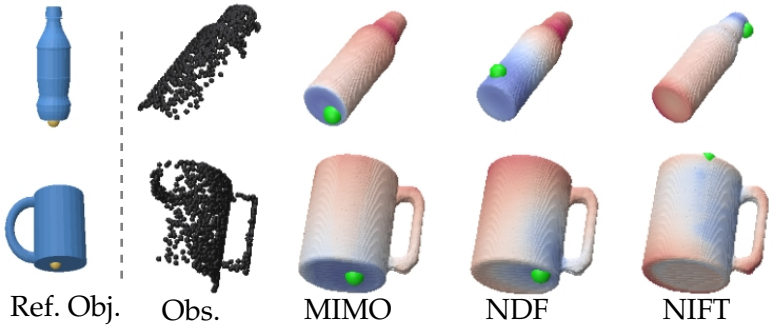
4.1.2. Neural Descriptors Derived from 3D Data

While neural descriptors derived from RGB images are effective for visible regions, they face significant limitations when it comes to modeling task constraints on hidden object parts due to the single viewpoint constraint. To overcome this, we can leverage three-dimensional representations, such as partially observed object point clouds, by training neural networks for object reconstruction and extracting 3D descriptors.

We introduced NDF and NIFT models in this direction in Section 3.1.2. However, they also exhibit limitation in capturing geometric details. As illustrated in Figure 4.6, when given a point selected on a reference object and a partially-observed categorical object, NDF produces imprecise point correspondences, while NIFT frequently fails to distinguish directional attributes, such as the up and down orientation of bottles or mugs. To create more informative descriptor fields, we propose Multi-feature Implicit Model (MIMO).

As depicted in Figure 4.7a, MIMO employs a Vector Neurons-PointNet (Deng et al., 2021) encoder $\varepsilon(\mathbf{P})$ that embeds the geometric information of point cloud $\mathbf{P}$ in an *equivariant* latent code. It also utilizes a partly shared Multi-Layer Perceptron (MLP) decoder with multiple branches to represent *SO(3)-invariant* spatial relations between a point $\mathbf{p}$ and point cloud $\mathbf{P}$.

The occupancy $\mathbf{\Phi}_{\text{occ}}$ (Mescheder et al., 2019) and signed distance $\mathbf{\Phi}_{\text{sdf}}$ (Park et al., 2019) branches enable MIMO to reconstruct object shapes. Specifically, given a fully- or partially-observed point cloud of an object, we extract the object mesh from the trained occupancy branch using the Multi-resolution IsoSurface Extraction algorithm Mescheder et al. (2019). Our experiments demonstrate that



Note: Reprinted from Cai et al. (2024). © 2024 IEEE.

Figure 4.6.: Comparison of point correspondence detection results of MIMO, NDF and NIFT. Given a point (•) on a reference object and partially-observed point clouds (•), we colorize the novel object mesh based on the L1 distance of point descriptors to the reference point, where blue means more similar, and marks the most similar points (•).

jointly training the signed distance and occupancy branches yields more precise shape reconstruction compared to training the occupancy branch alone.

Additionally, we introduce two novel feature branches: 1) the extended SCF (ESCF) branch $\mathbf{\Phi}_{\text{escf}}$, and 2) the closest distance direction (CDD) branch $\mathbf{\Phi}_{\text{cdd}}$. Unlike the SCF branch in NIFT, which only considers the power spectrum of each degree in the spherical harmonics expansion, our ESCF branch is directly supervised by the coefficients of spherical harmonics expansion across all orders and degrees. This approach enables ESCF to capture significantly finer geometric details.

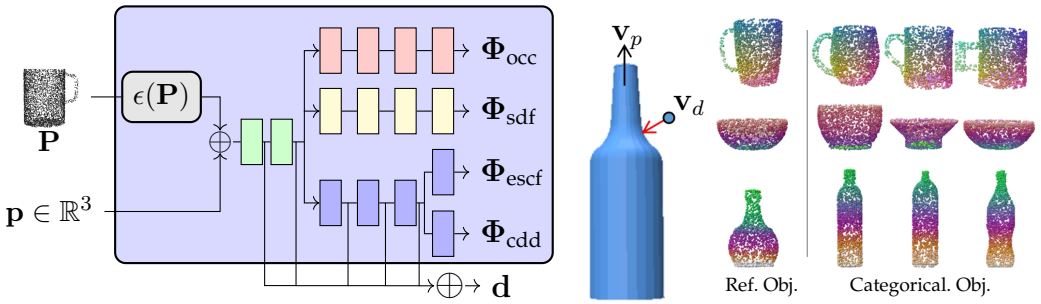
To further enhance the neural field's directional awareness, we introduce CDD, defined as the inner product of unit vectors $\mathbf{v}_d$ and $\mathbf{v}_p$. Here, $\mathbf{v}_d$ points from a point $\mathbf{p}$ to the closest point on the object, while $\mathbf{v}_p$ follows a chosen principal direction, such as pointing upward when the object is positioned vertically (see Figure 4.7b).

Similar to NDF, we concatenate the activation layers of the *partly-shared decoder* for $\mathbf{\Phi}_{\text{escf}}$ and $\mathbf{\Phi}_{\text{cdd}}$ to form the *point descriptor*:

$$\mathbf{d} = f_\theta^{\text{MIMO}}(\mathbf{p}|\mathbf{P}) : \mathbb{R}^3 \times \mathbb{R}^{3\times N_p} \to \mathbb{R}^D. \tag{4.18}$$

This descriptor space effectively measures geometric similarity (see Figure 4.7c). By training with four branches, our descriptor space becomes more informative in distinguishing fine geometric details.

In practice, we observed performance degradation in similarity measures when directly inferring point descriptors from noisy partially-observed point clouds



(a) Network Structure of MIMO. (b) CDD. (c) Similarity Measure.

Note: Adapted from Cai et al. (2024). © 2024 IEEE.

Figure 4.7.: The model architecture of Multi-feature Implicit Model (MIMO) and its application for similarity measure. (a) MIMO takes as input an object point cloud $\mathbf{P}$ and a point coordinate $\mathbf{p}$ and outputs multiple spatial features of $\mathbf{p}$ relative to $\mathbf{P}$, including occupancy $\mathbf{\Phi}_{\text{occ}}$, signed distance $\mathbf{\Phi}_{\text{sdf}}$, Extended Space Coverage Feature (ESCF) $\mathbf{\Phi}_{\text{escf}}$ and Closest Distance Direction (CDD) $\mathbf{\Phi}_{\text{cdd}}$. The concatenation of activation layers of the decoder for $\mathbf{\Phi}_{\text{escf}}$ and $\mathbf{\Phi}_{\text{cdd}}$ forms the point descriptor $\mathbf{d}$ of $\mathbf{p}$. (b) The CDD is represented as the inner product of two unit vectors $\mathbf{v}_p$ and $\mathbf{v}_d$. (c) The high-dimensional point descriptors of each reference object are reduced to a 3D space using PCA representing the RGB channels of the color map. Each point of other categorical object instances (at each row) is colorized according to the most similar point (smallest L1 distance in point descriptors) from the corresponding reference object at the same row.

$\mathbf{P}$. To address this challenge, we first reconstruct the mesh, then sample a point cloud $\mathbf{P}_r$ from it, which serves as input to MIMO for inferring the point descriptor $\mathbf{d} = f_\theta^{\text{MIMO}}(\mathbf{p}|\mathbf{P}_r)$. MIMO consistently outperforms both NDF and NIFT in finding accurate point correspondences, resulting in higher success rate in downstream manipulation tasks Cai et al. (2024). Comparison of correspondence detection performances with bottle and mug are shown in Figure 4.6.

Crucially, MIMO estimates neural descriptors for arbitrary points in 3D space around the object point cloud. This capability allows point-based task representations not only on the object surface but also in the surrounding space – a functionality that neural descriptors derived from RGB data cannot provide. Both types of neural descriptors presented in this thesis enable dense correspondence detection between categorical objects, facilitating object-centric task representations via the object canonical space, which we detail in the following section. The loss function design, and training details of the MIMO model are presented in Appendix A.4.

4.1.3. Object Canonical Space and Knowledge Transfer

In this thesis, the objective of object-centric task representation is achieved through the definition of an *object canonical space* – a correspondence-based representation framework that uses an exemplar object instance as a reference template for a given category. This space is defined by a 1) a *template object*, 2) a *set candidate points* 3) a *set of visual descriptors*, and 4) a *set of local frames*.

Canonical space using 2D descriptors

When using 2D descriptors, we assume a single viewpoint of an examplar object is provided. The canonical space derived from this viewpoint consists of:

1. A *template object*: An object intance of a category observed from a specific viewpoint, providing RGB, depth, mask information, and viewpoint augmentation;
2. A *set candidate points*: A set of densely sampled points within the masked region, lifted to 3D using the depth map;
3. A *set of visual descriptors*: the neural descriptors for each candidate point, extracted from visual foundation model or 3D neural descriptor fields; and
4. A *set of local frames*: Each reference frame is established using the neighborhood structure of each candidate point.

As illustrated in Figure 4.4, we select an exemplar kettle in reference image $\mathbf{A}_\mathrm{r}$ to construct the canonical space. We apply viewpoint augmentation and visual descriptor models to the augmented images. With a sampled set of reference pixels $\mathcal{X}_r$ on the exemplar object, we lift them to a 3D point set $\mathcal{P}_r$ and obtain their visual descriptors $\mathcal{D}_r$ using visual foundation models. Since $\mathcal{P}_r$ represents the observable shape of the object category, we refer to it as the *canonical shape*.

Based on this representation, we can employ SBBM and TPS methods to establish dense correspondences between the canonical space and a new kettle instance (see Figure 4.8-*top*). These correspondences enable keypoint and local pose mapping between categorical object instances, which is essential in knowledge transfer. For example, keypoints on the handle and spout can be precisely mapped to a target kettle (see Figure 4.8-*middle*). Furthermore, local frames anchored to these keypoints can be mapped to the target kettle by leveraging their local neighborhood structure (see Figure 4.8-*bottom*).

Formally, we define one canonical local frame $\mathcal{F}$ equal to identity centered at each reference point $\mathbf{p} \in \mathcal{P}_r$ of the canonical space. Each local frame is assigned

the Q closest reference points to its origin, whose positions in this local frame are denoted as the template neighborhood structure, $\mathbf{p}_q^*$. In other words, each local frame is parameterized by

$$\vartheta_{\mathcal{F}} = \left\{ \{\mathbf{d}_q\}_{q=1}^{Q}, \{\mathbf{p}_q^*\}_{q=1}^{Q} \right\}, \tag{4.19}$$

where $\mathbf{d}_q$ are the descriptors of the Q neighboring points.

These neighboring points are then used to detect the corresponding local frame on another instance of the same object category (see Figure 4.8-*bottom*). Specifically, the local frame $\mathcal{F}$ is detected by minimizing the mean squared displacement of the observed coordinates $\{\mathbf{p}_q\}_{q=1}^{Q}$ of the neighboring points with their reference values $\{\mathbf{p}_q^*\}_{q=1}^{Q}$:

$$\mathcal{F}^* = \arg\min_{\mathcal{F}} \sum_{q=1}^{Q} \left\| \mathbf{p}_q^* - \mathbf{p}_q \right\|^2. \tag{4.20}$$

The set of local frames on the reference object in canonical space is denoted as $\Theta = \left\{ \vartheta_{\mathcal{F}_j} : j \in [1, |\mathcal{P}_r|] \cap \mathbb{Z} \right\}$. The complete canonical space of an object therefore includes the sampled set of pixels, their 3D positions, neural descriptors, and parameters of local frames tied to them: $\mathsf{can}_2 = \{\mathcal{X}_r, \mathcal{P}_r, \mathcal{D}_r, \Theta\}$.

Canonical space using 3D descriptors

The correspondence detection using 3D descriptors differs in the neural fields $f_\theta^{\text{MIMO}}(\cdot \mid \mathbf{P})$ that is conditioned on the partially observed or reconstructed point cloud, which can be applied to arbitrary point in 3D space. Therefore, the canonical space can be constructed by densely sampling a set $\mathcal{P}_r$ of point from the 3D mesh of a template object of a category and computing their descriptors $\mathcal{D}_r$ using trained MIMO. Differently from the pose transfer of 2D case using point positions (Eqs. (4.19) and (4.20)), 3D pose transfer is achieved directly by minimizing the L1-distance between two pose descriptors (Eq. (3.4)), where each candidate local frame is parameterized by its neighboring points' descriptors $\vartheta_{\mathcal{F}} = \{\mathbf{d}_q\}_{q=1}^{Q}$. The canonical space can then be summarized as $\mathsf{can}_3 = \{\mathcal{P}_r, \mathcal{D}_r, \Theta\}$, where $\Theta = \left\{ \vartheta_{\mathcal{F}_j} : j \in [1, |\mathcal{P}_r|] \cap \mathbb{Z} \right\}$.

Summary

Overall, both types of canonical spaces enables four key capabilities:

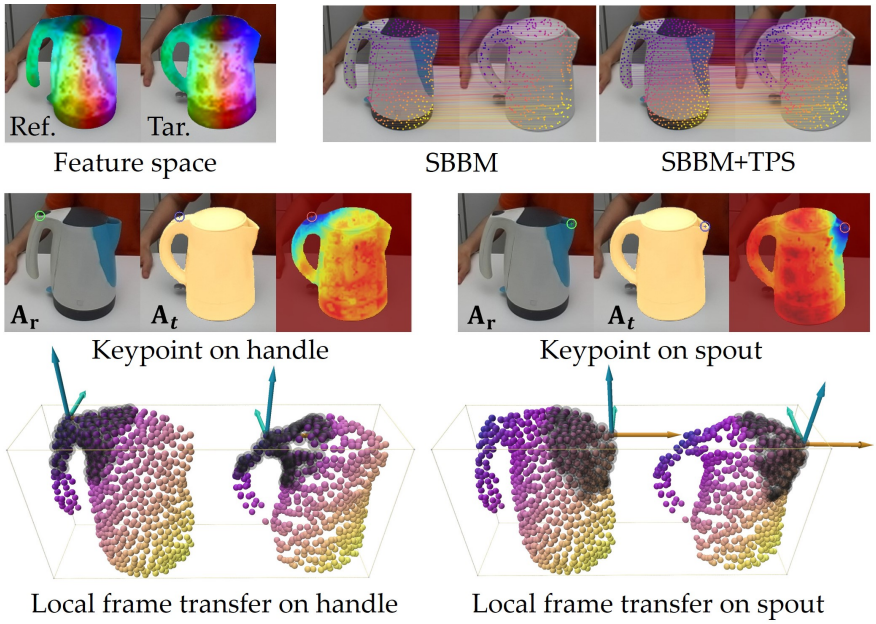


Figure 4.8.: Local frame detection and transfer leveraging object canonical space and 2D neural descriptors. Color gradient of correspondence pairs in SBBM and TPS are determined based on the distance of each pixel to the top left corner of the image, while the colors of the 3D points in the last row are the same as their 2D counterparts. Each local frame is determined by the closest 60 neighboring points (•).

1. Cross-instance correspondence detection within the same object category, applicable to both rigid and deformable objects;

2. Transfer of local pose information between different object instances;

3. Representation of tasks using keypoints and constraints in object-relative local coordinate frames; and

4. Generalization of manipulation skills across varying object geometries.

The canonical space serves as a critical bridge between perception and action, where object-specific properties are anchored to semantically consistent reference points rather than to absolute spatial coordinates or idealized object models. Task constraints are encoded relative to these semantically consistent local frames, creating a representation that maintains invariance to instance-level geometric variations and viewpoint changes, while preserving category-level semantic structure.

4.1.4. Perception Pipeline

Human demonstration videos of manipulation tasks present significant challenges for task learning due to inconsistent viewpoints, object variations, and occlusions. In the previous section, we proposed an object canonical space and neural descriptor-based correspondence detection approach that ensures object-centric representation independent of viewpoints and invariant to instance-level geometric variations. This section details the integration of the proposed method, the state-of-the-art keypoint tracking algorithms, and human pose estimation into a robust perception pipeline for preprocessing human demonstration videos.

We record[1] N human demonstration videos $\mathcal{V} = \{V_n\}_{n=1}^{N}$ of a unimanual manipulation task in 3-dimensional task space, with each video containing a sequence of RGB and depth images $V_n = \left\{ \{\mathbf{A}(t)\}_{t=1}^{T}, \{\mathbf{D}(t)\}_{t=1}^{T} \right\}$, where T is the time horizon[2]. We assume that all demonstrations contain the motion segment of interest and that the list of object categories $\mathcal{O} = \{O_i\}_{i=1}^{N_o}$ involved in this task is known, where N_o is the number of objects. For the visual perception pipeline relying on 2D neural descriptors, we additionally assume that all task-relevant keypoints are located in regions on the object surface that remain visible to the imitator and are not occluded in the first image frame.

Our data preprocessing comprises the following steps: 1) Object canonical space construction; 2) Dense correspondence detection; 3) Dense keypoints tracking and occlusion-aware warping; 4) Human hand pose estimation and tracking; 5) Local pose trajectory estimation; and 6) Object role assignment. Note that this section focuses on the perception pipeline using 2D neural descriptors; while the 3D descriptors will be used in Section 6.3 for pose transfer tasks.

Constructing object canonical space

We begin by taking the first image frame of each video and detecting object instances using GroundingDINO Liu et al. (2024), an open-vocabulary object detection model that produces bounding boxes based on language prompts containing object category information. These detected bounding boxes serve as prompt for the Segment Anything Model (SAM, Kirillov et al., 2023) to estimate binary masks corresponding to each object.

[1] Details of hardware setup and the video format are presented in Appendix A.1.
[2] We resample the demonstration videos to have an equal number of image frames.

We then select detected objects from a random demonstration to construct object canonical spaces following the steps presented in Section 4.1.3, as only one reference object per category is required. The canonical space of each object O_i includes sampled pixels, their corresponding 3D points (i. e., *canonical shape*), and descriptors $\{\mathcal{X}_{r,i}, \mathcal{P}_{r,i}, \mathcal{D}_{r,i}, \Theta_i\}$. Any point in the sets of the sampled reference points across all objects is considered a *candidate keypoint*, while the local frame tied to it is a *candidate local frame*. All candidate points in 2D and 3D for a given task are denoted by:

$$\mathcal{X}_c = \bigcup_{i \in [1, N_o] \cap \mathbb{Z}} \mathcal{X}_{r,i}, \quad \mathcal{P}_c = \bigcup_{i \in [1, N_o] \cap \mathbb{Z}} \mathcal{P}_{r,i}. \tag{4.21}$$

We denote the total number of candidate points as $N_c = |\mathcal{P}_c|$.

Furthermore, we define the *spatial scale* $\varphi_i \in \mathbb{R}$ as the maximum distance between any pair of candidates on the canonical shape of each object (see details of mathematical formulation in Appendix B.1). This property will be used in Section 4.2.1 to determine object-independent thresholds.

Dense correspondence detection

Categorical objects detected in other demonstrations are treated as target objects. For these objects, we estimate dense correspondence pairs using SBBM and TPS, which are further used to estimate local frames tied to each detected correspondence point on the target object (see Figure 4.8).

For each n-th demonstration at the first image frame ($t = 1$), we obtain a binary mask image $\mathbf{M}_i^n(t)$ for each object O_i, along with dense correspondence points – represented as a set of 2D pixels $\mathcal{X}_i^n(t)$ and a set of 3D points $\mathcal{P}_i^n(t)$ – corresponding to each object's canonical space $\{\mathcal{X}_{r,i}, \mathcal{P}_{r,i}, \mathcal{D}_{r,i}, \Theta_i\}$. As illustrated in Figure 4.9, kettles in nine demonstrations are detected, and point correspondences between them are established via the canonical space. Simultaneously, a set of local frames $\Theta_{O_i}^n(t)$ tied to local neighborhoods of each object are estimated according to Eq. (4.20). Two example local frames on the spout and handle of the kettle are shown in Figure 4.9. Importantly, these local frames enable us to align demonstrations recorded from different viewpoints to a common perspective, facilitating geometric constraint extraction and keypoint identification, as will be discussed in Section 4.2.

Dense correspondence detection on the first image frame of each demonstration provides $\{\mathbf{M}_i^n(t), \mathcal{X}_i^n(t), \mathcal{P}_i^n(t), \Theta_i^n(t) \mid t = 1, i = [1, N_o] \cap \mathbb{Z}\}$, which will subse-

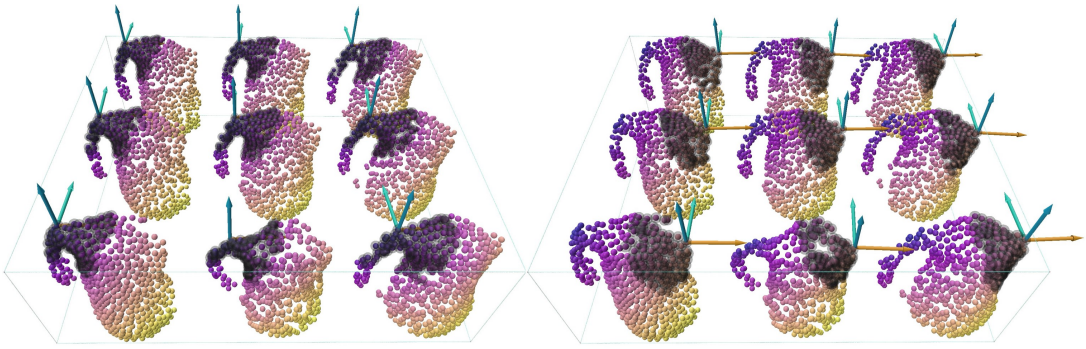


Figure 4.9.: Local frame detection and transfer on kettle objects across nine demonstrations. Two example local frames on the spout and handle are shown. The object instance in the bottom left corner represents the canonical space.

quently be used for tracking object masks and keypoints in Section 4.1.4 and estimating local pose trajectories in Section 4.1.4.

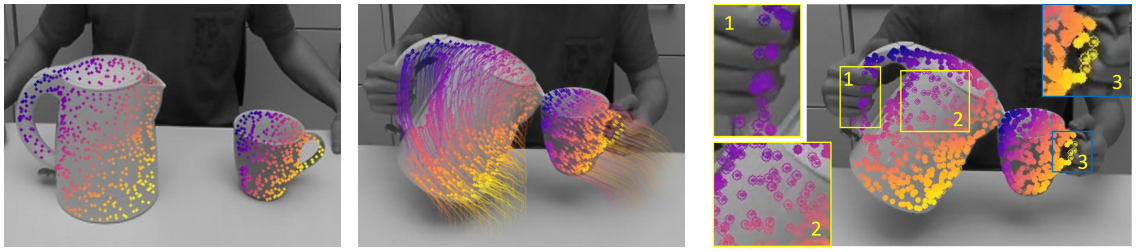
Dense keypoints tracking and occlusion-aware warping

To track the motion of all candidates and objects throughout the demonstration, we employ Segment and Track Anything (SAM-Track, Cheng et al., 2023) for object mask tracking based on the initial mask image $\mathrm{M}_i^n(t)$ of each object O_i at $t = 1$. We integrate the Spatial Tracker model (Xiao et al., 2024) for dense point tracking. As shown in Figure 4.10, spatial tracker takes the 2D dense correspondence points of all object categories at the first image frame as query points:

$$\mathcal{X}^n_{\text{query}} = \bigcup_{i \in [1, N_o] \cap \mathbb{Z}} \mathcal{X}^n_i(t), \quad t = 1, n \in [1, N] \cap \mathbb{Z}, \quad |\mathcal{X}^n_{\text{query}}| = N_c \tag{4.22}$$

and tracks these pixels over time for a given demonstration video. The resulting pixel trajectories must be lifted to 3D trajectories of candidate points before constraint extraction. Visible pixels can be projected to 3D using the Lift operator and the depth image, while occluded pixels cannot be naively lifted to 3D as their corresponding depth values are incorrect. To overcome this issue, we need to estimate the occlusion status of each candidate point and determine their 3D positions based on their observable neighborhoods.

While Spatial Tracker can robustly handle occlusion between objects or between hand and object in 2D image space, its estimated occlusion status is not consistently reliable. As shown in Figure 4.10c, the human left hand occludes the cup handle in area 3, where the occlusion status is accurately estimated. How-



(a) Query points (b) Tracks (c) Occlusion

Figure 4.10.: Keypoints tracking with occlusion-aware Spatial Tracker. (a) Starting from the dense correspondence point sets of all objects at the first image frame ($t = 1$) as query point, (b) Spatial Tracker trackers these pixels even when they are under occlusion. (c) The occluded pixels are displayed as hollow circles while observable ones with filled circles.

ever, in area 1, the occlusion status of pixels on the kettle's handle occluded by the human right hand is not detected, while occluded pixels in area 2 are mostly false positives. To address this, we propose an alternative method to estimate occlusion status based on object binary masks and depth maps, which is both computationally efficient and more reliable than Spatial Tracker. Detailed methodology is provided in Appendix A.5.

We then apply Thin-plate-spline (TPS) algorithm for each object O_i recursively, starting from the 3D positions $\mathcal{P}_i^n(t)$ at $t = 1$. At each time step t, we use the 3D positions of all visible points on the object as control points to warp the 3D positions of occluded points based on their positions at the previous time step $t - 1$. By applying these tracking and warping steps to all N demonstration videos, we obtain 3D trajectories of densely sampled candidate points on objects, denoted by $\mathcal{T}_c = \{\boldsymbol{\tau}_h\}_h^{N_c}$, where $\boldsymbol{\tau}_h \in \mathbb{R}^{3\times N\times T}$.

These trajectories are represented in the camera frame. To eliminate noise from keypoint detection and tracking algorithms, we utilize Savitzky-Golay (SG) filters (Press and Teukolsky, 1990). The SG filters also allow us to obtain velocity trajectories as the first-order time derivatives of position trajectories. For simplicity, we denote the smoothed trajectories with the same symbols $\mathcal{T}$ and $\boldsymbol{\tau}$ throughout the remaining content.

Human hand pose estimation and tracking

In addition to candidate points on objects, KVIL requires keypoints of human hands and handedness information (i. e., left/right hand labels). In natural visual demonstrations, human hands often experience heavy (self-)occlusion in several image frames, causing methods like MediaPipe Lugaresi et al. (2019)

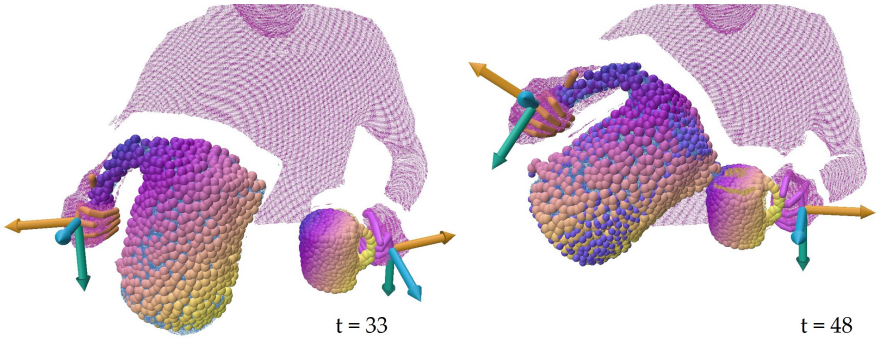


Figure 4.11.: Hand pose estimation and tracking using InterWild.

and RTMPose Jiang et al. (2023) to fail. We found that InterWild (Moon, 2023) robustly estimates human hand poses, providing representations of hands as 3D meshes, 3D keypoints, and handedness information, which enables mapping different sub-tasks to the robot's hands. While InterWild's estimation of hand 3D positions based on monocular images and estimated camera parameters is not accurate or temporally consistent, its 2D keypoints on images are highly accurate. Therefore, we re-base the hand mesh to the most probable visible keypoint of the hand lifted to 3D using the depth map, where the probability of each keypoint is determined by object-hand mask overlay and pre-defined priority. The human binary masks are also tracked with SAM-Track. We empirically observed that our framework outperforms other models such as MediaPipe, RTMPose, and OSX Lin et al. (2023) when hands are under heavy (self-)occlusion.

InterWild uses the MANO model (Romero et al., 2017) of human hands, which contains 21 keypoints for each hand. In total, we obtain trajectories of hand keypoints $\{\boldsymbol{\tau}_j\}_{j=1}^{N_h}$, where $\boldsymbol{\tau}_j \in \mathbb{R}^{3\times N\times T}$ and $N_h = 21$ is the number of keypoints on a hand for unimanual tasks in this chapter, while $N_h = 42$ for bimanual manipulation in Chapter 5. Similar to object trajectories, we apply SG filters to smooth the hand trajectories and compute their velocities. Since both hands and objects in human demonstrations exhibit the same format of point data, we treat the human hand as a special object type in the remainder of this thesis. As shown in Figure 4.11, the hand skeleton connecting the 21 keypoints at two timesteps is visualized for the pouring water task. The hand-object interaction is precisely captured in this point-based dataset, facilitating fine-grained task representation, including motion segmentation and grasp detection Section 6.1.1, task-oriented grasp modeling Section 6.3.

Local pose trajectory estimation

As explained in Section 4.1.3, we can estimate local frames tied to each candidate using their closest Q neighbors. Having obtained trajectories of all candidate points throughout the demonstrations, we can apply Eq. (4.20) to each candidate point at each time step to derive a set of local pose trajectories $\mathcal{T}_{\mathcal{F}} = \{\boldsymbol{\tau}_{\mathcal{F},h}\}_{h}^{N_c}$, where $\boldsymbol{\tau}_{\mathcal{F},h} \in \mathbb{R}^{4\times4\times N\times T}$ when a local frame is represented as a homogeneous transformation matrix. We define one local frame per hand attached to the middle finger MCP keypoint (Romero et al., 2017), the orientation is defined with z-axis pointing from palm to finger, x axis go through palm and y axis from thumb to pinky finger, as detailed in Appendix A.2. Their trajectories $\boldsymbol{\tau}_{\mathcal{F},h}$ are also estimated for each demonstration, where $h \in \{\mathsf{Left}, \mathsf{Right}\}$.

Object role assignment

Geometric constraints alone are insufficient to fully represent a task. For example, one constraint in a pouring task is kettle-cup alignment, which could be achieved by moving the cup toward a static kettle. However, pouring specifically requires motion of the kettle. Uni-KVIL addresses this issue by considering the role of objects for the task at hand. We detect object motion saliency using the candidate trajectories similarly to Muhlig et al. (2009b) to determine object roles $\mathcal{R} = \{\gamma_i\}_{i=1}^{N_o}$, where $\gamma_i \in \{\mathsf{master}, \mathsf{slave}\}$.

The master object O_m is defined as the object with the lowest average variance of candidates' trajectories, while other objects are designated as slave object O_s. Uni-KVIL accounts for these object roles by extracting local frames only on the master object and extracting keypoints only on the slave objects.

Summary

In summary, the preprocessed data from the N demonstration videos contains:

1. the *canonical spaces* of objects involved in the task, including sampled pixels, their 3D positions (i. e., *canonical shape*), neural descriptors, and parameters of candidate local frames tied to them $\{\{\mathcal{X}_{r,i}, \mathcal{P}_{r,i}, \mathcal{D}_{r,i}, \Theta_i\}\}_{i=1}^{N_o}$;
2. the set of *object spatial scales* $\Phi = \{\varphi_i\}_{i=1}^{N_o}$;
3. trajectories of *candidate points* and *candidate local frames* across all demonstrations $\mathcal{T}_c, \mathcal{T}_{\mathcal{F}}$; and
4. the set of object roles $\mathcal{R} = \{\gamma_i \mid \gamma_i \in \{\mathsf{master}, \mathsf{slave}\}\}_{i=1}^{N_o}$.

This structure is illustrated in the overview diagram (see Figure 4.12). Based on the assigned roles of objects as either master or slave, the candidates (see Eq. (4.21)) can be divided into two subsets: candidate points on master objects and those on slave objects. This division is similarly applied to their corresponding descriptors, position trajectories, and pose trajectories:

$$\mathcal{P}_c = \mathcal{P}_m \cup \mathcal{P}_s, \qquad \mathcal{P}_d = \mathcal{D}_m \cup \mathcal{D}_s \tag{4.23}$$

$$\mathcal{T}_c = \mathcal{T}_m \cup \mathcal{T}_s, \qquad \mathcal{T}_{\mathcal{F}} = \mathcal{T}_{\mathcal{F},m} \cup \mathcal{T}_{\mathcal{F},s}. \tag{4.24}$$

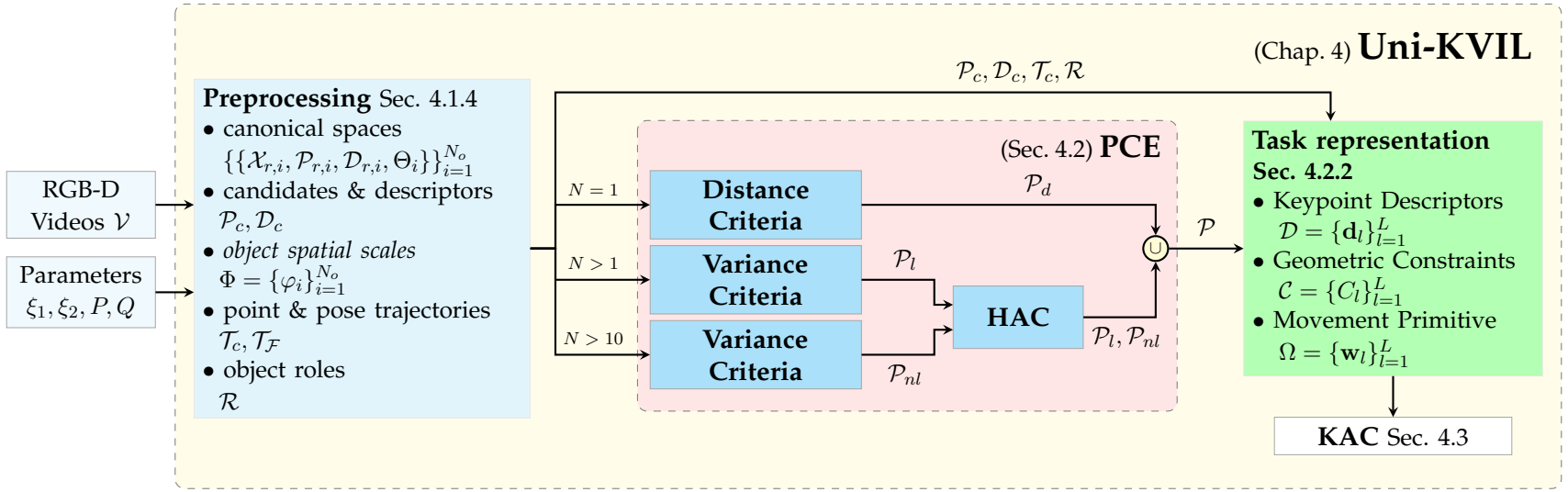
4.2. Extracting Keypoint-based Task Representation

Given the preprocessed data, our objective is to simultaneously extract a set $\mathcal{P}$ of keypoints and a corresponding set $\mathcal{C} = \{C_l\}_{l=1}^{L}$ of geometric constraints. We define six elementary geometric constraints in a 3D Cartesian space, namely point-to-point (p2p), point-to-line (p2l), point-to-plane (p2P), point-to-curve (p2c), point-to-surface (p2S), and pose constraints, as illustrated in Figure 4.13.

To determine which candidate points are subject to linear constraints – such as p2p, p2l, and p2P – we analyze the principal components of each candidate point's position across multiple demonstrations using *PCA* (Halko et al., 2011). This analysis is based on the explained variance of each principal component (see Section 4.2.1). Conversely, nonlinear geometric constraints (p2c and p2S) are estimated through the iterative *PME* method (see Section 3.2). Complex constraints such as *colinearity*, *coplanarity*, *parallelism*, and *perpendicularity* emerge from combinations of these fundamental constraint types.

Additionally, a distribution of hand grasp poses or object poses associated with specific functional parts can be modeled using Gaussian Mixture Models (GMMs). However, Uni-KVIL primarily focuses on learning constraints for action segments corresponding to object manipulation, without addressing automatic grasping. We, therefore, discuss the modeling of pose constraints and task-oriented grasping separately in Section 6.3.

To ensure robust constraint estimation, the criteria for Principal Constraints Estimation (PCE) adapt to the number of demonstrations. For single-demonstration scenarios (i. e., one-shot imitation learning), learning generalizable skills is inherently limited due to the lack of variability. Thus, we employ heuristic



Note: Adapted from Gao et al. (2023). © 2023 IEEE.

Figure 4.12.: Overview of Uni-KVIL's architecture. After preprocessing the demonstration videos, Uni-KVIL jointly extracts a set of sparse keypoints, a set of keypoint-based geometric constraints, and a set of movement primitive parameters that fully represent the task. The robot then leverages the proposed keypoint-based admittance controller to reproduce the task in novel scenes.

distance-based criteria to fully constrain the pose of objects (see Section 4.2.1). In contrast, when multiple demonstrations are available (i. e., few-shot imitation learning), keypoints and geometric constraints are determined based on the spatial distribution and variability of candidate points across demonstrations (see Section 4.2.1). Nonlinear constraints are incorporated only when a sufficient number of demonstrations are available.

4.2.1. Principal Constraints Estimation

We now introduce the criteria for estimating linear and nonlinear geometric constraints on principal manifolds.

Distance criteria for a single demonstration

A single demonstration ($N = 1$) does not provide sufficient variation to learn generalizable skills. Consequently, we constrain the pose of objects by assuming rigidity and extracting three keypoints for each slave object to fully define its pose in 3D space.

We map the trajectories of all candidate points on the slave objects into candidate local frames $\mathcal{F}_j(t)$ on the master object at each time step t, aligning demonstrations to a common viewpoint, where $\mathcal{F}_j \in \mathcal{T}_{\mathcal{F},m}$ is the j^{th} candidate local frame on the master object. Examples of viewpoint alignment are discussed in Section 4.4.2. We denote these transformed trajectories as $\tilde{\boldsymbol{\tau}}_j^k(t) \in \mathbb{R}^{3\times N}$, the positions of the k^{th} candidate point on a slave object represented in the j^{th} candidate local frame at timestep t across N demonstrations. We use this variable to denote the *candidate positions* in the remainder of this thesis.

Empirical observations suggest that, in daily manipulation tasks, the closest point $\mathbf{k}_1$ on the slave object to the master object is crucial for maintaining contact or avoiding collisions. The farthest point $\mathbf{k}_2$, in combination with $\mathbf{k}_1$, helps define the object's pose. Based on these heuristics, we extract the local frame $\mathcal{F}^*(t)$ from all candidates that is, on average, closest to all candidates on the slave objects. We then identify two keypoints, $\mathbf{k}_1$ and $\mathbf{k}_2$, as the closest and farthest candidates from $\mathcal{F}^*(t)$, respectively. For 3D tasks, an additional keypoint $\mathbf{k}_3$ is selected as the farthest from both $\mathbf{k}_1$ and $\mathbf{k}_2$.

These three keypoints fully define the slave object's pose and are subject to p2p constraints. For each slave object, we obtain a set $\mathcal{P}_d = \{\mathbf{k}_l\}_{l=1}^3$ of keypoints and their associated geometric constraints $\mathcal{C} = \{C_l\}_{l=1}^3$, where each constraint

C_l specifies a p2p constraint on a "0-dimensional" principal manifold at time t within the selected local frame:

$$C_l = \left\{ \mathcal{M}_{\text{point}}(\mathbf{k}_l), t, \theta_{\mathcal{F}^*(t)}, \text{p2p} \right\}, l \in \{1, 2, 3\}, \tag{4.25}$$

where $\theta_{\mathcal{F}^*(t)}$ is the parameterization of the selected local frame.

To fully determine the pose of the slave object, the three keypoints are subject to linear p2p constraints. For each slave object, we finally obtain a set $\mathcal{P}_d = \{k_l\}_{l=1}^D$ of keypoints and the corresponding geometric constraints $\mathcal{C} = \{C_l\}_{l=1}^D$, where $C_l = \{\mathcal{M}_{\text{point}}(k_l), t, \vartheta_{\mathcal{F}^*(t)}, \text{p2p}\}$ defines a p2p constraint on a 0-dimensional principal manifold $\mathcal{M}_{\text{point}}$ on point k_l at time t in the local frame parameterized by $\vartheta_{\mathcal{F}^*}$.

Notably, the local frame parameters (Eq. (4.19)) encode the position profiles of neighboring points and their neural descriptors. These descriptors enable detecting neighborhood points on a new instance of the master object via dense correspondence detection (Section 4.1.1). This information, in turn, is used in Eq. (4.20) to estimate the pose of the local frame in the new scene, thereby enabling the transfer of keypoint constraints to novel environments and supporting an object-centric task representation with category-level generalization.

Variance criteria for linear constraints

When several demonstrations ($N > 1$) are available, we leverage their variability to estimate linear constraints beyond p2p. To do so, we obtain the candidate positions $\tilde{\boldsymbol{\tau}}_j^k(t)$ in the canonical local frames $\mathcal{F}_j(t)$ at time t as described in Section 4.2.1. We then compute the explained variance $\boldsymbol{\nu}_j^k(t)$ of each candidate's position at timestep t across multiple demonstrations using PCA.

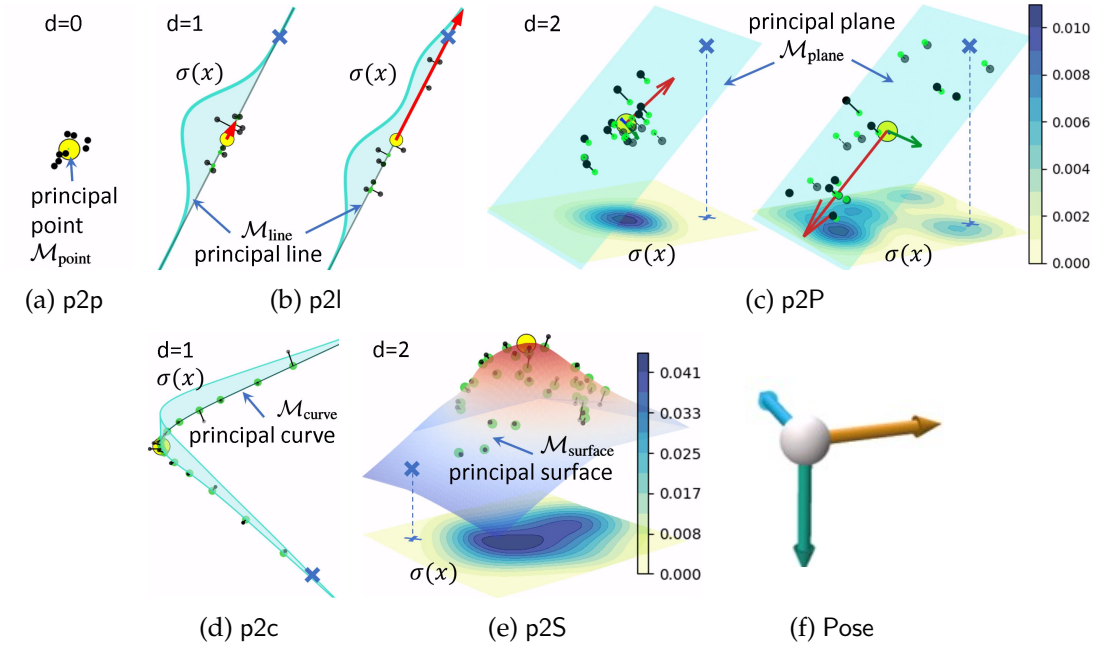
$$\boldsymbol{\nu}_j^k(t) = \mathbb{V}_{\text{PCA}}[\tilde{\boldsymbol{\tau}}_j^k(t)] \in \mathbb{R}^3, \quad \tilde{\boldsymbol{\tau}}_j^k(t) \in \mathbb{R}^{3\times N}. \tag{4.26}$$

The *spatial variability* $\boldsymbol{\eta}_j^k(t)$ is then defined as

$$\boldsymbol{\eta}_j^k(t) = \left(\boldsymbol{\nu}_j^k(t)\right)^{1/2} / \tilde{\varphi}_i, \tag{4.27}$$

with $\tilde{\varphi}_i$ the spatial scale of the slave object O_i to which the k^{th} candidate belongs.

In contrast to the explained variance, spatial variability removes dependencies on the object size. This allows us to empirically define two object-agnostic lower



Note: Adapted from Gao et al. (2023). © 2023 IEEE.

Figure 4.13.: Six types of geometric constraints. The constraints are obtained from candidate points (•) from N demonstrations. The density function $\sigma(\mathbf{z}), \mathbf{z} \in \mathbb{R}^d$ is estimated from the projections (•) of the candidate positions on the d-dimensional principal manifold. We also depict the mean $\mathbf{p}_m$ (•), the spatial variability (→, →) on the principal manifold along the principal components, the stress vector $\boldsymbol{s}$ (—), and examples of extrapolation of the keypoints (×) on the manifolds.

and upper thresholds ξ_1, ξ_2 to identify appropriate linear geometric constraints based on the computed spatial variability.

Candidates are classified into constraints based on the following conditions: 1) Low spatial variability in all components ($\eta^k_{j,1}(t) < \xi_1$) suggests a fixed-point constraint (p2p); 2) variability along only one principal component ($\eta^k_{j,2}(t) < \xi_1$, $\eta^k_{j,1}(t) > \xi_2$) suggests a line constraint (p2l); and 3) variability along two principal components ($\eta^k_{j,3}(t) < \xi_1$, $\eta^k_{j,2}(t) > \xi_2$) suggests a plane constraint (p2P).

Any spatial variability $\boldsymbol{\eta}^k_j(t)$ satisfying the above conditions indicates the joint selection of the k^{th} candidate point, the j^{th} candidate local frame at time step t. All candidates selected as such form a set $\mathcal{P}_l$ of keypoints subject to linear geometric constraints.

Note that, due to the fact that two distinct points define a line and three non-collinear points define a plane, we learn p2l constraints when $N > 2$ and of p2P constraints when $N > 3$.

Variance criteria for nonlinear constraints

Linear constraints may not sufficiently capture certain task constraints, such as the different inclination angles of the kettle across different pouring demonstrations (see Figure 1.2a). In such cases, nonlinear constraints (p2c, p2S) are estimated using PME.

Candidate points on slave objects that do not satisfy any linear constraints are considered potential candidates for nonlinear constraints. In our case, we replace the random D-dimensional vector $\mathbf{p}$ in (3.9) with the candidate point's positions $\tilde{\boldsymbol{\tau}}_j^k(t)$ at time step t across multiple demonstrations, so that the PME loss in Eq. (3.9) becomes

$$\mathcal{L}(f, \pi_d) = \mathbb{E} \left\| \tilde{\boldsymbol{\tau}}_j^k(t) - f(\pi_d(\tilde{\boldsymbol{\tau}}_j^k(t))) \right\|^2 + \lambda \|\kappa_f\|^2, \quad k \in [1, |\mathcal{P}_s|] \cap \mathbb{Z}.$$

After obtaining the projection index π_d from PME, we compute the projections of candidates onto the manifold, i. e.,

$$\hat{\boldsymbol{\tau}}_j^k(t) = \pi_d(\tilde{\boldsymbol{\tau}}_j^k(t)), \quad \hat{\boldsymbol{\tau}}_j^k(t) \in \mathbb{R}^{d \times N}. \tag{4.28}$$

Then, analogously to Section 4.2.1, we define the explained variance $\nu_{j,\parallel}^k(t)$ in the tangential direction of the principal manifold as the variance of the projections:

$$\nu_{j,\parallel}^k(t) = \mathbb{V}[\|\hat{\boldsymbol{\tau}}_j^k(t)\|] \in \mathbb{R}. \tag{4.29}$$

The explained variance $\nu_{j,\perp}^k(t)$ in the orthogonal direction corresponds to the variance of the length of the stress vectors

$$\nu_{j,\perp}^k(t) = \mathbb{V}[\|\mathbf{s}\|], \quad \mathbf{s} = \tilde{\boldsymbol{\tau}}_j^k(t) - f(\hat{\boldsymbol{\tau}}_j^k(t)), \tag{4.30}$$

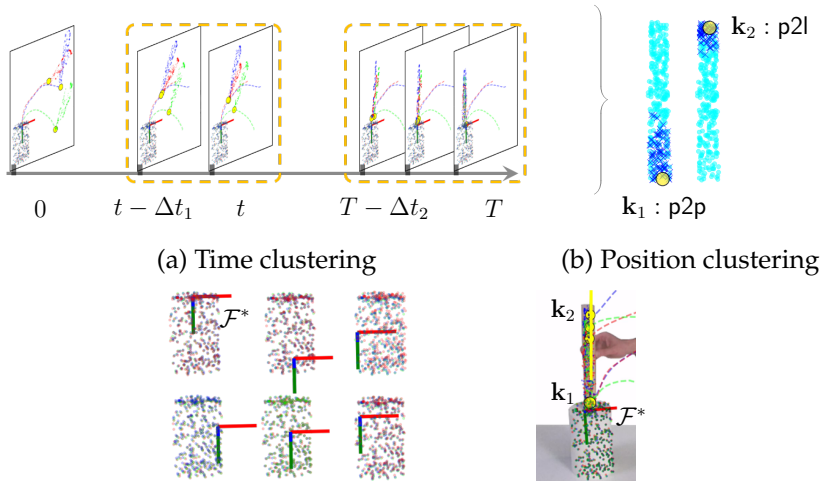
where $f(\cdot)$ is the reconstruction function of PME and $f(\hat{\boldsymbol{\tau}}_j^k(t)) \in \mathbb{R}^{3 \times N}$ are the orthogonal projection points of $\tilde{\boldsymbol{\tau}}_j^k(t)$.

Similar to the linear case, the spatial variability is defined as

$$\eta_{j,z}^k(t) = \sqrt{\nu_{j,z}^k(t)} / \tilde{\varphi}_i, \quad z \in \{\perp, \parallel\}. \tag{4.31}$$

The set of keypoints subject to nonlinear geometric constraints is determined by:

$$\mathcal{P}_{nl} = \left\{ \mathbf{p}_k \mid \nu_{j,\perp}^k(t) < \xi_1, \nu_{j,\parallel}^k(t) > \xi_2, \mathbf{p}_k \in \mathcal{P}_s, k \in [1, |\mathcal{P}_s|] \cap \mathbb{Z} \right\}. \tag{4.32}$$



(a) Time clustering (b) Position clustering

(c) Equivalent local frames $\mathcal{F}$ (d) Insertion

Note: Adapted from Gao et al. (2023). © 2023 IEEE.

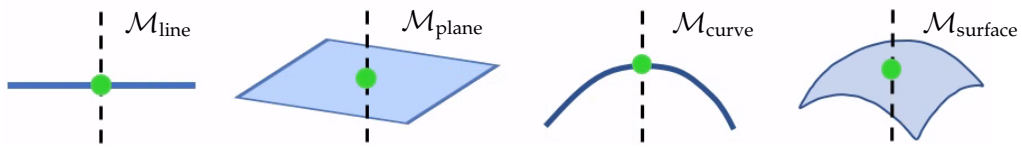
Figure 4.14.: Hierarchical agglomerative clustering for inserting a stick into a paper roll (see also Section 4.4 for the task description). (a) The selected candidates are first clustered in time (- -) to identify the adjacent timesteps. (b) The candidates (×) in each time cluster (e. g., here in the last time cluster $[T - \Delta t_2, T]$) are then clustered based on their positions in the canonical shape (●) of the stick for each constraint (here p2p and p2l). For each position cluster, the keypoint (●) with the lowest variability is finally selected. (c) Since the paper roll has no shape variation, i.e., all canonical local frames are equivalent, the closest frame $\mathcal{F}^*$ to the selected keypoints is selected. (d) Final task representation (see also Figure 4.22).

The type of the geometric constraints is determined by the intrinsic dimension d of the learned principal manifold, i. e., $d = 1$ and $d = 2$ indicate a p2c and a p2S constraint, respectively. Notice that, in order to guarantee their reliable estimation, nonlinear constraints are considered only when enough demonstrations (e. g., $N \geq 8$) are available.

Hierarchical Agglomerative Clustering (HAC)

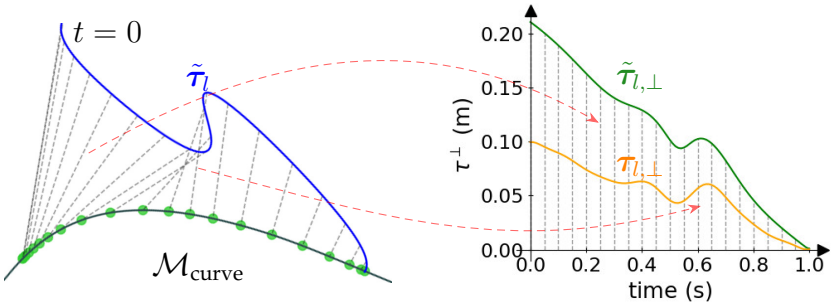
As described in Section 4.2.1, each selected candidate keypoint $\mathbf{k}$ in the sets of linear and nonlinear constraints, $\mathcal{P}_l$ and $\mathcal{P}_{nl}$, corresponds to a specific time step t and local frame $\mathcal{F}_j$. Redundancy may arise due to adjacent timesteps, neighboring keypoints, or equivalent local frames.

To address this, we first apply Hierarchical Agglomerative Clustering (HAC) to group keypoints in time, identifying adjacent timesteps. Figure 4.14 illustrates an example of HAC applied to an insertion task. Next, we cluster keypoints



Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Figure 4.15.: Orthogonal direction (- -) to the principal manifolds.



Note: Adapted from Gao et al. (2023). © 2023 IEEE.

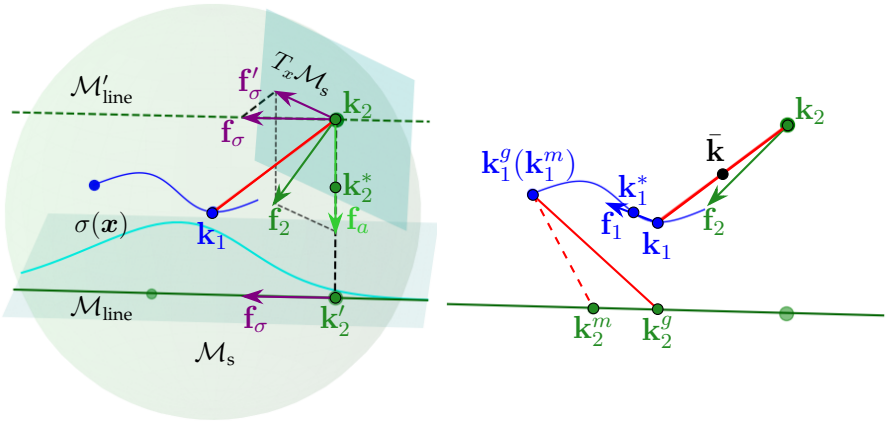
Figure 4.16.: Projected and reproduced trajectories using a VMP for a p2c constraint. The demonstrated trajectory $\tilde{\tau}_l$ (—) is projected at each time step in the orthogonal direction (- -) of the principal manifold $\mathcal{M}_{\text{curve}}$. The projected trajectory $\tilde{\tau}_{l,\perp}$ (—) is used to train movement primitives, which is then used to reproduce trajectories, e. g., $\tau_{l,\perp}$ (—) with a new start position at $0.1\,\text{m}$ and goal position at $0\,\text{m}$. The arrows (-→) mark the corresponding projected trajectory between the 3D and 2D plots at two timesteps.

within each time cluster based on their positions in the canonical shape of the slave object, thus grouping neighboring keypoints. Redundancy is resolved by selecting the keypoint with the lowest variability within each position cluster, ensuring robustness against sensor and correspondence detection noise. If multiple constraints exist for a selected keypoint at a given time step in different local frames, we choose the local frame closest to the keypoint on average.

In summary, the proposed PCE extracts a sparse set of L keypoints, given by $\mathcal{P} = \mathcal{P}_d \cup \mathcal{P}_l \cup \mathcal{P}_{nl}$, along with their associated (non)linear constraints, $\mathcal{C} = \{C_l\}_{l=1}^{L}$, which are used to represent the task as described next.

4.2.2. Extraction of the Complete Task Representation

The keypoints and associated constraints extracted in Section 4.2 provide an understanding of the demonstrated task. However, a control policy is necessary for task reproduction. We propose modeling the observed keypoint trajectories



Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

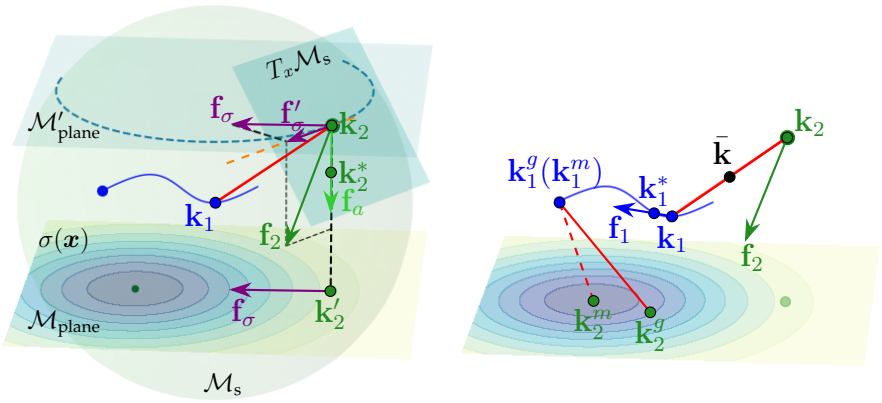
Figure 4.17.: Illustration of the attraction and density forces when the keypoints $\mathbf{k}_1$ and $\mathbf{k}_2$ are subject to p2p and p2l constraints, respectively. **Left:** The approach force $\mathbf{f}_a$ of $\mathbf{k}_2$ is computed by the virtual spring-damper system between the attractor $\mathbf{k}_2^*$ and $\mathbf{k}_2$. The density force $\mathbf{f}_\sigma$ is then projected onto the tangent space of the sphere at $\mathbf{k}_2$. The control force of $\mathbf{k}_2$ is the combination of the attraction and the projected density force. **Right:** $\mathbf{k}_1$ is controlled by the attraction force $\mathbf{f}_1$ following the attractor $\mathbf{k}_1^*$ and the VMP (—) to reach the target $\mathbf{k}_1^g$, which coincides with the demonstrated target $\mathbf{k}_1^m$. Note that the target $\mathbf{k}_2^g$ of $\mathbf{k}_2$ does not coincide with $\mathbf{k}_2^m$ due to object shape variation, i.e., the distance (—) between $\mathbf{k}_1$ and $\mathbf{k}_2$ during reproduction is longer than for the demonstration (- -).

using VMPs (Zhou et al., 2019). These VMPs are trained from an object-centric perspective and adhere to the constraints estimated via PCE.

Specifically, for each keypoint subject to a p2p constraint, we train a VMP on its observed trajectory $\tilde{\boldsymbol{\tau}}_l$ within the corresponding local frame $\mathcal{F}_j$ from time step 1 to the extracted time step t, i. e., $\tilde{\boldsymbol{\tau}}_l = (\boldsymbol{\tau}_j^k(1), \ldots, \boldsymbol{\tau}_j^k(t))^\mathsf{T}$, where k and l indicates that the k^{th} candidate point in the dense set $\mathcal{P}_s$ is selected as the l^{th} keypoint in the sparse set $\mathcal{P}$.

For constraints with intrinsic dimension $d > 0$ (e. g., p2l, p2P, p2c, p2S), fulfillment requires that the keypoint reach the corresponding principal manifold at the end of the control period. While the exact location on the manifold does not affect constraint satisfaction, it influences the similarity between the demonstrated and reproduced object poses. Thus, we decompose keypoint control into orthogonal and tangential components relative to the principal manifold (Figure 4.15).

The orthogonal motion ensures adherence to the constraint by guiding the keypoint onto the manifold, while the tangential motion regulates object pose similarity between demonstrations and reproduction. Since slave object's shape



Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Figure 4.18.: Illustration of the attraction and density forces when $\mathbf{k}_2$ is subject to a p2P constraint. In contrast to Figure 4.17, the density force $\mathbf{f}_\sigma$ is projected onto the intersection line (- -) of the shifted principal manifold $\mathcal{M}'_{\text{plane}}$ and the tangent space. Legend as in Figure 4.17.

variations may affect target keypoint positions on the principal manifold, we train the VMP only on the orthogonal component of the trajectory, $\tilde{\boldsymbol{\tau}}_{l,\perp}$.

Notably, the principal manifold reliably predicts the expansion trend of the geometric constraints even to regions in space far away from the training data distribution. This allows Uni-KVIL to extrapolate keypoints target positions along the manifold, a unique property that enhances the generalization capability, which has not yet been demonstrated in the literature.

Figure 4.16 presents an example of projected and reproduced trajectories under a p2c constraint. During reproduction, at each time step, we determine the 3D target position of a keypoint by adding the VMP-generated offset in the orthogonal direction of the principal manifold, controlling the keypoints toward its orthogonal projection on the principal manifold (e. g., $\mathbf{k}_2^*$ in Figure 4.17 and Figure 4.18). Setting the VMP goal to zero ensures constraint satisfaction at the trajectory's end. The keypoint's motion along both directions is regulated by the keypoint-based admittance controller introduced in the next section.

In conclusion, Uni-KVIL's final task representation consists of a set of keypoints, characterized by their descriptors $\mathcal{D} = \{\mathbf{d}_l\}_{l=1}^L$, their geometric constraints $\mathcal{C} = \{C_l\}_{l=1}^L$, and movement primitives encoded via the weight set $\Omega = \{\mathbf{w}_l\}_{l=1}^L$.

4.3. Keypoints-based Admittance Controller

After learning the representation of a given task from demonstrations, our goal is to enable robots to reproduce this task by interacting with objects such that their keypoints fulfill the learned constrained. This requires bridging the gap between Uni-KVIL's task representation and real-time robot controllers. To address this challenge, we propose a *Keypoint-based Admittance Controller* (KAC), which effectively: 1) accommodates variable numbers of keypoints across different tasks; 2) balances constraint fulfillment with extrapolation of keypoint target positions on their learned principal manifolds; 3) resolves potential interference between different types of geometric constraints. Notably, capabilities 2) and 3) are essential for handling significant object shape variations during task reproduction.

The KAC associates each keypoint in $\mathcal{P}$ with a virtual spring-damper system, whose attractor is computed via the corresponding VMPs (see Section 4.3.1). As detailed in Section 4.3.4, the accumulated *attraction forces* from all keypoints' spring-damper systems serve as the task-space force command for the robot. This design enables the KAC to handle varying numbers of keypoints across different tasks. Regarding capability 2), extrapolation of keypoint target positions under `p2p` constraints is prohibited. For non-`p2p` constraints, the controller decomposes control forces into orthogonal and tangential directions relative to the learned principal manifolds (see Section 4.2.2). Consequently, keypoints approach the principal manifolds following motion profiles learned from the projected trajectories in orthogonal directions.

The control force generated by each keypoint's virtual spring-damper system maintains orthogonality to the principal manifold throughout execution, ensuring keypoints satisfy the corresponding geometric constraints when the VMP execution concludes. While this approach allows for the extrapolation of keypoint target positions on the principal manifold, it does not consider the distance between demonstrated[3] and extrapolated targets[4]. Therefore, we propose balancing extrapolation and regulation by estimating the density function of demonstrated targets on the principal manifolds, as described in Section 4.3.2. This density information is then utilized to compute an additional force – the *density force* – that guides each keypoint toward demonstrated targets.

[3]Demonstrated targets refer to the keypoint target positions on the principal manifold obtained from demonstrations.

[4]Extrapolated targets are keypoint target positions sampled on the principal manifold outside the demonstrated distribution.

Finally, we obtain capability 3) by assigning different priorities to different types of geometric constraints, as described in Section 4.3.3. The following subsections detail each component of the KAC.

4.3.1. Attraction force

Given the learned task representation and a new image frame $\mathbf{A}_\mathrm{t}$ for task reproduction, we identify the keypoints representing the task. Specifically, we estimate the correspondence point of each keypoint $\mathbf{k}_l$ on $\mathbf{A}_\mathrm{t}$ using their visual descriptors $\mathbf{d}_l$ and dense correspondence detection (see Section 4.1.1). These positions are then mapped to robot local frame $\mathcal{F}_r$.

The attractor $\mathbf{k}_l^*$ of the virtual spring-damper system at each time step– i. e., the keypoint target position – is computed for each keypoint by the corresponding VMPs projected onto $\mathcal{F}_r$. The attraction force generated by the virtual spring-damper system is then computed as:

$$\mathbf{f}_{a,l} = \bar{\mathbf{K}}_p(\mathbf{k}_l^* - \mathbf{k}_l) + \bar{\mathbf{K}}_d(\dot{\mathbf{k}}_l^* - \dot{\mathbf{k}}_l), \tag{4.33}$$

where $\bar{\mathbf{K}}_p, \bar{\mathbf{K}}_d$ are diagonal stiffness and damping matrices, respectively, and $\dot{\mathbf{k}}_l$ and $\dot{\mathbf{k}}_l^*$ are the velocity of $\mathbf{k}_l$ and $\mathbf{k}_l^*$, respectively.

As explained in Section 4.2.2, for non-p2p constraints, the VMPs are trained on trajectories projected in directions orthogonal to the principal manifold. Therefore, the learned VMPs and resulting attraction forces reproduce the demonstrated motion patterns in the orthogonal direction. This means that the final positions of keypoints can be extrapolated anywhere on the principal manifolds to accommodate object shape variations and other geometric constraints. However, without considering the demonstrated targets on the principal manifold, we risk losing critical information about successful task execution or specific execution styles. We capture this information using *kernel density estimation* and provide additional *density forces* to guide keypoints toward demonstrated targets.

4.3.2. Density force

For a non-p2p constraint, we project the demonstrated keypoint positions $\tilde{\boldsymbol{\tau}}_l(t)$ at the extracted time step t onto the corresponding d-dimensional principal manifold using the learned projection index π_d, yielding $\hat{\mathbf{k}}_l^m = \pi_d(\tilde{\boldsymbol{\tau}}_l(t)) \in \mathbb{R}^{N\times d}$.

Since at time step t the l^{th} keypoint should fulfill the geometric constraints, we interpret $\hat{\mathbf{k}}_l^m$ as its demonstrated target positions on the manifold.

We then estimate the density function $\sigma(\mathbf{x})$ of the keypoint target positions from $\hat{\mathbf{k}}_l^m$ using kernel density estimation Pedregosa et al. (2011) with SE kernels. Examples of estimated density functions for p2l, p2P, p2c, and p2S constraints are illustrated in Figures 4.13b to 4.13e. This density function indicates the probability of a keypoint target position on the corresponding principal manifold given the demonstrated target positions. In other words, the density function quantifies the confidence level when extrapolating keypoint target positions to new locations on the principal manifold, which may occur due to object shape variations during reproduction.

Examples of extrapolated keypoint target positions ($\times$) during reproduction are shown in Figures 4.13b to 4.13e. Notice that the target position in Figure 4.13b-*left* has a lower probability (i. e., lower extrapolation confidence) than the one in Figure 4.13b-*right* due to its greater distance from demonstrated target positions. This illustrates that keypoint control must not only fulfill geometric constraints but also position keypoints as close as possible to demonstrated targets on the constraints.

Therefore, in addition to the attraction force $\mathbf{f}_{a,l}$ that ensures geometric constraint fulfillment, we define a *density force* $\mathbf{f}_{\sigma,l}$ to guide keypoints toward regions with higher probability. First, we define the driving force $\mathbf{f}_{\sigma,1}$ computed from the density gradient $\nabla\sigma(\mathbf{x})$ as:

$$\mathbf{f}_{\sigma,1} = g_1 \cdot f(\nabla\sigma(x)). \tag{4.34}$$

For regions where $\mathbf{f}_{\sigma,1}$ is insufficient to move keypoints effectively, we define a minimal driving force $\mathbf{f}_{\sigma,2}$ as:

$$\mathbf{f}_{\sigma,2} = g_2 \cdot f(\frac{\mathbf{k}^m - \mathbf{k}'_l}{\|\mathbf{k}^m - \mathbf{k}'_l\|_2}), \tag{4.35}$$

where $\mathbf{k}'_l = \pi_d(\mathbf{k}_l) \in \mathbb{R}^d$ is the projection of the keypoint onto the principal manifold, $f(\cdot)$ is the reconstruction function (see Section 4.2.1), and g_1 and g_2 are force scaling parameters. Note that $\mathbf{f}_{\sigma,2}$ points directly to the mean of the demonstrated targets $\hat{\mathbf{k}}_l^m$ on the principal manifold, i. e., $\mathbf{k}^m = \text{avg}(\hat{\mathbf{k}}_l^m)$.

The density force is then defined as the maximum of $\mathbf{f}_{\sigma,1}$ and $\mathbf{f}_{\sigma,2}$:

$$\mathbf{f}_{\sigma,l} = \arg\max_{\mathbf{f}} \|\mathbf{f}\|_2, \quad \mathbf{f} \in \{\mathbf{f}_{\sigma,1}, \mathbf{f}_{\sigma,2}\}. \tag{4.36}$$

Examples of these density forces $\mathbf{f}_\sigma$ for p2l and p2P constraints are illustrated in Figures 4.17 and 4.18.

4.3.3. Priority

When accommodating large object shape variations, controlling a p2l constraint with the same priority as a p2p constraint may lead to violations of the latter. To mitigate such interference, we assign higher priority to p2p constraints compared to other constraint types.

For clarity, we use Figures 4.17 and 4.18 to explain this concept. Figure 4.17 shows the case of two constraints: p2p for $\mathbf{k}_1$ and p2l for $\mathbf{k}_2$, while in Figure 4.18, $\mathbf{k}_2$ is subject to a p2P constraint. In both scenarios, we construct a sphere centered at $\mathbf{k}_1$ with radius $\|\mathbf{k}_2 - \mathbf{k}_1\|$ and define the tangent space of the sphere at $\mathbf{k}_2$ as the plane formed by all lines tangent to the sphere at $\mathbf{k}_2$.

Figures 4.17 and 4.18 also depict the corresponding principal manifolds $\mathcal{M}_{\text{line}}$ and $\mathcal{M}_{\text{plane}}$ shifted in parallel to pass through $\mathbf{k}_2$ as $\mathcal{M}'_{\text{line}}$ and $\mathcal{M}'_{\text{plane}}$. Assuming solid connections (—) between $\mathbf{k}_1$ and $\mathbf{k}_2$, strong density forces $\mathbf{f}_\sigma$ applied to $\mathbf{k}_2$ would drag $\mathbf{k}_1$ in the same direction. This could violate the p2p constraint for $\mathbf{k}_1$ and potentially cause collisions if $\mathbf{k}_1$ is near the master object.

To reduce such interference when the principal manifold is a line $\mathcal{M}_{\text{line}}$, we project $\mathbf{f}_\sigma$ onto the tangent space $T_x\mathcal{M}_{\text{s}}$ of the sphere $\mathcal{M}_{\text{s}}$, ensuring that $\mathbf{k}_1$'s position remains unaffected by the projected density force $\mathbf{f}'\sigma$ (see Figure 4.17). Similarly, when the principal manifold is a plane (Figure 4.18), we project $\mathbf{f}_\sigma$ onto the intersection between the tangent space and the shifted principal plane $\mathcal{M}'_{\text{plane}}$. This approach also applies to nonlinear constraints (p2c and p2S), for which we use a linear approximation at each time step.

In summary, the density force ensures that the reproduced task resembles the demonstrations on the principal manifold, while the priority mechanism reduces interference with p2p constraints while preserving the extrapolation capability. The decomposition of control forces into attraction forces (Section 4.3.1) and density forces (Sections 4.3.2 and 4.3.3) is crucial for balancing reproduction fidelity and extrapolation capability.

In practice, we empirically tune the stiffness and damping gains of the virtual spring-damper systems to achieve optimal tracking accuracy and control stability. One-shot IL represents a special case (see Section 4.2.1) where the learned task representation comprises three keypoints subject to p2p constraints. In this scenario, no density force is required, and constraint priorities are defined

as $\mathrm{Pri}_1 > \mathrm{Pri}_2 > \mathrm{Pri}_3$, since $\mathbf{k}_1$ typically represents the contact point between objects. To ensure higher control precision for $\mathbf{k}_1$, we set the stiffness gains of the three keypoints to $\bar{\mathbf{K}}_{p,1} = 5\bar{\mathbf{K}}_{p,2} = 10\bar{\mathbf{K}}_{p,3}$ and the respective damping gains to $\bar{\mathbf{K}}_{d,l} = 2\bar{\mathbf{K}}_{p,l}^{1/2}$, where $l \in [1, 3]$, ensuring critically damped behavior for control stability.

4.3.4. Admittance controller

The KAC computes the robot arm control command by combining attraction forces $\mathbf{f}_a$ and projected density forces $\mathbf{f}'\sigma$ for all keypoints. First, we compute the control force for each keypoint as $\mathbf{f}_l = \mathbf{f}_{a,l} + \mathbf{f}'_{\sigma,l}$ and define a virtual tool-center-point (TCP) $\bar{\mathbf{k}} = \sum_{l=1}^{L} \mathbf{k}_l / L$ as the mean of all keypoint positions (see Figure 4.17-*right*, Figure 4.18-*right*).

The virtual TCP is driven by a virtual force and torque:

$$\mathbf{f}_f = \sum_{l=1}^{L} \mathbf{f}_l \quad \text{and} \quad \mathbf{f}_\tau = \sum_{l=1}^{L} (\mathbf{k}_l - \bar{\mathbf{k}}) \times \mathbf{f}_l$$

where $\times$ denotes the vector cross product.

The total control force $\mathbf{f}_v = [\mathbf{f}_f^\mathsf{T}, \mathbf{f}_\tau^\mathsf{T}]^\mathsf{T}$ is applied to the robot end-effector (i.e., the humanoid hand) to calculate the virtual acceleration:

$$\ddot{\mathbf{x}}_v = \tilde{\mathbf{K}}_p(\mathbf{x}_0 - \mathbf{x}_v) - \tilde{\mathbf{K}}_d\dot{\mathbf{x}}_v - \tilde{\mathbf{K}}_m\mathbf{f}_v,$$

where $\mathbf{x}_0, \mathbf{x}_v$ are the initial and virtual poses of the robot end-effector, $\dot{\mathbf{x}}_v$ is its virtual velocity, and $\tilde{\mathbf{K}}_m, \tilde{\mathbf{K}}_d$, and $\tilde{\mathbf{K}}_p$ are the inertia, damping, and stiffness factors, respectively.

The robot is controlled using a task space inverse dynamics controller, whose task space control force $\mathbf{f}_m$ is calculated as:

$$\mathbf{f}_m = \mathbf{K}_p(\mathbf{x}_v - \mathbf{x}) + \mathbf{K}_d(\dot{\mathbf{x}}_v - \dot{\mathbf{x}}) + \mathbf{h}_c,$$

where $\mathbf{x}$, $\dot{\mathbf{x}}$ are the current end-effector pose and velocity, $\mathbf{K}_d$, and $\mathbf{K}_p$ are the damping and stiffness factors of the impedance controller, respectively, and $\mathbf{h}_c$ represents the Coriolis and gravitational force in the task space.

4.4. Evaluation

We evaluate our approach across five daily manipulation tasks involving diverse geometric constraints and various categorical objects (see Figure 4.19). The tasks include: press button (PB, Figure 4.20), fetch tissue (FT, Figure 4.21), insert sticks into a paper roll (IS, Figure 4.22), pour water (PW, Figure 4.23), hang hat on a rack (HH, Figure 4.24), and clean table with a dustpan and a brush (CT, Figure 4.25).

For all experiments, the number of candidate points on each slave object is set to $P_i = 300$ for objects and $P_i = 21$ for hands, while the number of neighboring points around each candidate local frame on the master object is $Q = 50$ for objects and $Q = 10$ for hands. We utilized 20 kernels for the VMPs. Thresholds ξ_1 and ξ_2 are empirically selected, and controller gains are tuned based for our humanoid robot.

Our evaluation first examines Uni-KVIL's ability to extract generalizable task representations given varying numbers of demonstrations (see Section 4.4.2 for one-shot and Section 4.4.2 for few-shot visual imitation learning, as well as Figures 4.20 to 4.25). We demonstrate how variations in objects' pose and shape contribute to efficient extraction of generalizable task representations and evaluate Uni-KVIL's ability to reproduce corresponding tasks learned from different numbers of demonstrations.

We discuss challenges arising from scarce demonstrations in Section 4.4.2 and show how these challenges are resolved through additional demonstrations. We summarize the minimum number of demonstrations required to learn a generalizable representation for each task. Finally, we evaluate the proposed KAC in terms of control accuracy, precision, and success rate in Section 4.4.3. More evaluation results can be found in the accompanying website[5] of the published paper (Gao et al., 2023).

4.4.1. Evaluation Protocols

We record human demonstration videos as detailed in Appendix A.1. For tasks involving categorical objects, we distinguish between the object instances used for the demonstrations, and for the reproductions. When only one instance of a specific object category was available, we used it for demonstrations and reproductions.

[5] https://sites.google.com/view/k-vil

Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

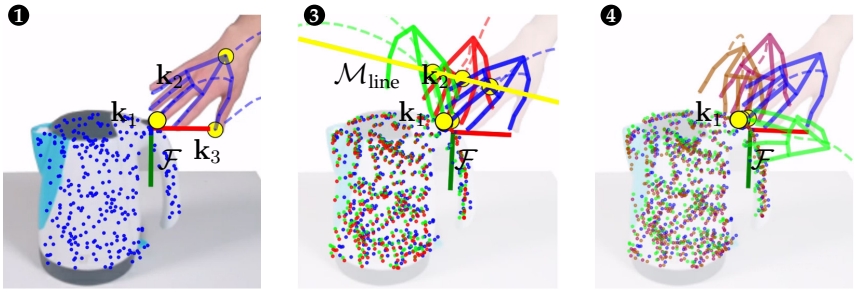
Figure 4.19.: Objects used in our paper include (a) tissue boxes, (b) teacups, (c) a rack and a hat, (d) kettles, (e) a paper roll, (f) sticks, (g) dustpans and brushes. Note that the rack can be assembled with stick #6-10 to have multiple shape variations.

We define a set of *extraction tasks* $\mathcal{T}_E = \{\mathsf{PB}, \mathsf{FT}, \mathsf{PW}, \mathsf{HH}, \mathsf{IS}, \mathsf{CT}\}$ for which we evaluate Uni-KVIL's ability to extract generalizable task representations given $N \in \{1, 3, 4, 5, 11\}$ demonstrations. For clarity, we evaluate only the representations of the last time cluster, i. e., the goal configuration of each task when $t = T$. This can be easily extended to other time clusters if necessary.

We then define a set of *reproduction tasks* $\mathcal{T}_R = \{\mathsf{Task}\,\textcircled{N} : \mathsf{Task} \in \mathcal{T}_E\}$, each representing the reproduction of Task by ARMAR-6 using the task representation extracted from N demonstrations.

To evaluate reproduction and adaptation of learned task representations in new cluttered scenes, we arbitrarily perturb the scene before each execution trial. Specifically, we place involved objects and robot hands in different locations within the workspace and camera view. We use the image frame captured by the robot right before the execution to parameterize the learned task representation, including identifying keypoints, detecting local frames, configuring geometric constraints, and generating keypoint motion trajectories using learned VMPs.

We consider a task learned from N demonstrations and from a third-person view to be *generalizable* when it can be successfully reproduced by the robot with



(a) p2p: $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ (b) p2p: $\mathbf{k}_1$, p2l: $\mathbf{k}_2$ (c) p2p: $\mathbf{k}_1$

Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

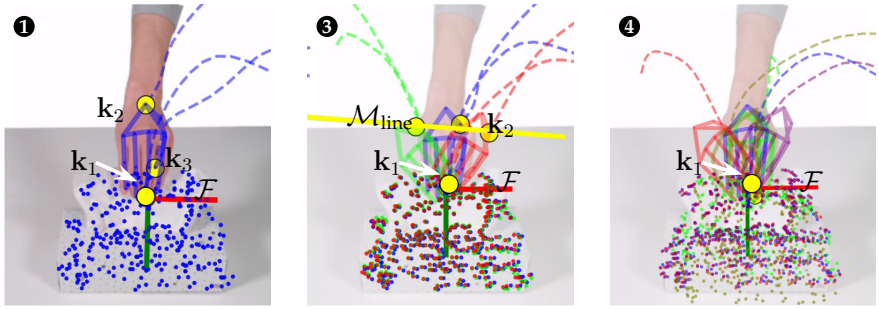
Figure 4.20.: Press a button on the kettle to open the lid with variations in hand orientation. The candidate points (colored points) and the skeleton of hands (colored line segments) are overlain on the objects at time step T. Different colors denote different trials. The keypoints $\mathbf{k}_l$ (○) are extracted from different number Ⓝ of demonstrations in each subfigure. The demonstrated trajectories (- -), the local frames $\mathcal{F}$, and the estimated principal manifolds (—) are also depicted. Notice that the demonstrations were provided from different viewpoints, although they are here represented aligned to the local frame $\mathcal{F}$ for a better illustration of the extracted keypoints and constraints (for viewpoint mismatch, see Figure 4.25.)

categorical objects in new cluttered scenes. Below, we describe specifications for each task in terms of demonstration collection and successful robot reproduction. Table 4.7 lists all considered tasks.

Press Button (PB): A human demonstrates opening a kettle lid by pressing its button with the middle finger's tip using either hand. We used kettle #5 in Figure 4.19d for this task. The demonstrations include different hand poses approaching the button (see Figure 4.20). For robot reproduction, we designed fixed maps between human hand keypoints and robot hand keypoints. Successful reproduction requires the robot to reach the button with its fingertip within $5\,\text{mm}$ and open the lid by slightly closing the finger to press the button.

Fetch Tissue (FT): A human demonstrates fetching tissue from two different tissue boxes (#3 and #4 in Figure 4.19a) with various hand poses (see Figure 4.21). Tissue boxes #4-10 were used for reproduction. Successful reproduction requires the robot to grasp the tissue and pull it out with a predefined pulling action. Although shape variations between demonstration boxes #3-4 are subtle, box #10 introduces significant shape variations for reproduction testing.

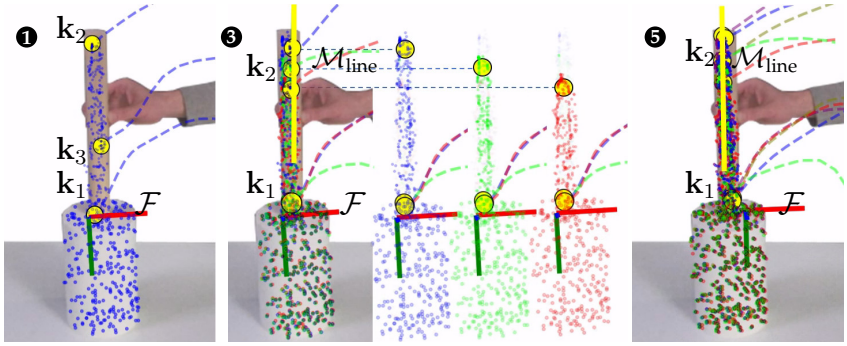
Insert Stick (IS): Sticks #2-4 in Figure 4.19f and paper roll in Figure 4.19e were used for human demonstrations (see Figure 4.22). We did not insert sticks fully into the paper roll to be able to observe the aligning status of the stick to its line constraints. While pose variations were not considered, shape variations



(a) p2p: $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ (b) p2p: $\mathbf{k}_1$, p2l: $\mathbf{k}_2$ (c) p2p: $\mathbf{k}_1$

Note: Adapted from Gao et al. (2023). © 2023 IEEE.

Figure 4.21.: Fetch tissue with variations in hand orientation. Legend as in Figure 4.20.



(a) p2p: $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ (b) p2p: $\mathbf{k}_1$, p2l: $\mathbf{k}_2$ (c) p2p: $\mathbf{k}_1$, p2l: $\mathbf{k}_2$

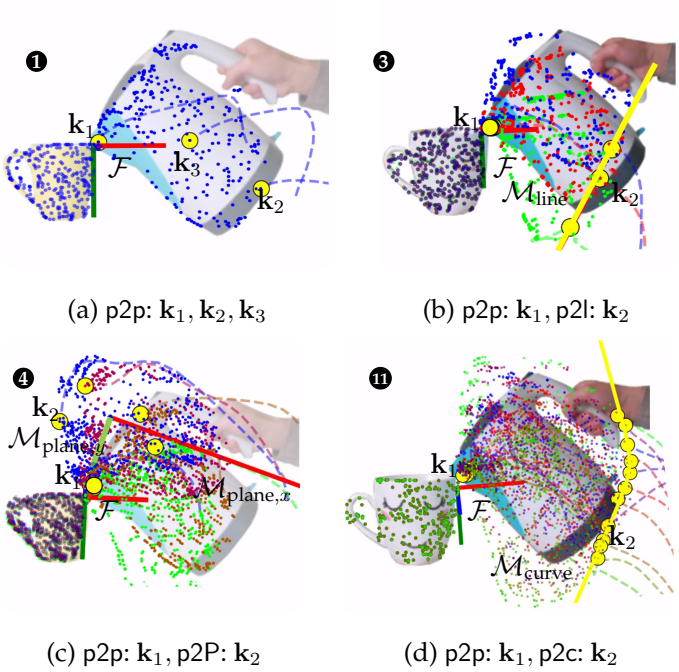
Note: Adapted from Gao et al. (2023). © 2023 IEEE.

Figure 4.22.: Approach the insertion position to insert sticks with 3 length variations into a paper roll. Legend as Figure 4.20.

were introduced through sticks of different lengths and thicknesses. For reproduction, sticks were initially tilted between $-30°$ to $80°$, extending beyond the demonstration range ($0°$ to $45°$). Successful reproduction requires placing the stick's lower tip directly above the paper roll's center hole without collision to the paper roll during the execution.

Pour Water (PW): A human demonstrated pouring with kettle #5 in Figure 4.19d and teacups #1 and #3 in Figure 4.19b, incorporating teacup shape variations and kettle pose variations (see Figure 4.23). Teacups #1-4 and all kettles were used for reproduction. Successful reproduction requires aligning the kettle spout above the teacup rim and tilting the kettle appropriately.

Hang Hat (HH): The rack was assembled with different-length sticks (#6-10 in Figure 4.19f). Racks assembled with sticks #7-8 were used for demonstrations, while sticks #6-10 were used for reproduction. Specifically, stick #7 was used



(a) p2p: $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ (b) p2p: $\mathbf{k}_1$, p2l: $\mathbf{k}_2$

(c) p2p: $\mathbf{k}_1$, p2P: $\mathbf{k}_2$ (d) p2p: $\mathbf{k}_1$, p2c: $\mathbf{k}_2$

Note: Adapted from Gao et al. (2023). © 2023 IEEE.

Figure 4.23.: Pouring task with variations in the shape of cups and the orientation of the kettle. The principal plane in (b) is represented by orthogonal vectors $\mathcal{M}_{\text{plane},x}$ (—) and $\mathcal{M}_{\text{plane},y}$ (—). Other legends as in Figure 4.20.

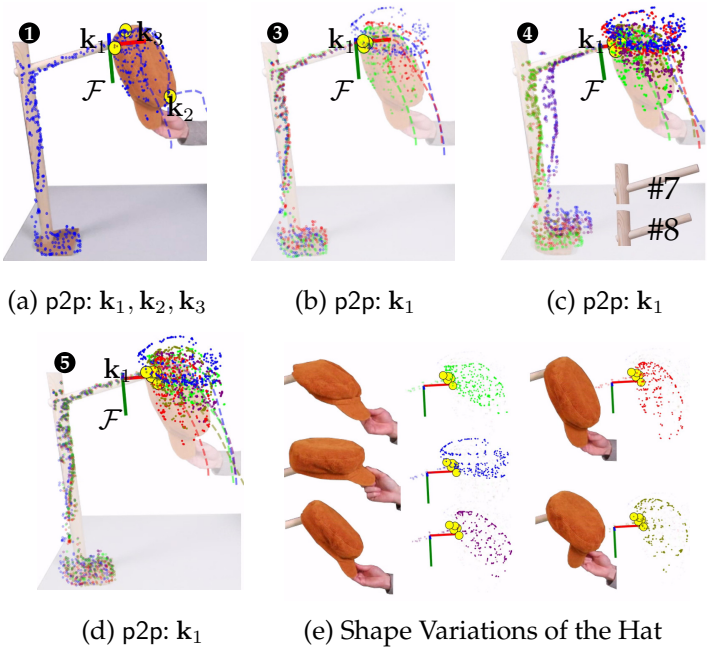
in Figures 4.24a, 4.24b and 4.24d and sticks #7-8 in Figure 4.24c. Successful reproduction occurs when the hat rim is placed on the stick tip regardless of stick length and initial hat pose.

Clean Table (CT): Dustpans #2-3 and brushes #3-4 in Figure 4.19g were used for demonstrations, while dustpan #1 and brushes #1-2 were used for reproduction. The task is successful when the brush head aligns parallel above the dustpan edge.

4.4.2. Evaluation of Uni-KVIL's Task Representation

As discussed in Section 4.2, Uni-KVIL's task representation can be acquired from one or a few demonstration videos based on distance and variance criteria. Therefore, the number of demonstrations and variations in object poses and shapes play essential roles.

In this evaluation, we address three key questions: (i) How do task representations learned from different numbers of demonstrations affect reproduction performance? What are the limitations of representations learned from scarce



(a) p2p: $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ (b) p2p: $\mathbf{k}_1$ (c) p2p: $\mathbf{k}_1$

(d) p2p: $\mathbf{k}_1$ (e) Shape Variations of the Hat

Note: Adapted from Gao et al. (2023). © 2023 IEEE.

Figure 4.24.: Hang a hat on a rack. Note that there are only slightly deformations of the hat in (a)-(c) and relatively more obvious deformations in (d) (see (e)), while pose variations in the hats are considered in all cases. The racks in (c) have two shape variations. Legend as Figure 4.20.

demonstrations? (ii) How many demonstrations are required to learn generalizable task representations? (iii) How do shape and pose variations contribute to successful extraction of task representations?

We evaluate Uni-KVIL in one-shot and few-shot imitation learning setups in Section 4.4.2 and Section 4.4.2, respectively, and answer these questions in Section 4.4.2.

One-shot Imitation Learning

With a single demonstration, Uni-KVIL learns a task representation based on the distance criteria presented in Section 4.2.1, resulting in a set of three linear p2p constraints.

Task extraction: We first learned the insertion task from a single demonstration involving stick #4 (Figure 4.19f) and a paper roll. Here, the master object is the paper roll, and the slave object is the stick. As shown in Figure 4.22a, Uni-KVIL extracts three keypoints subject to p2p constraints on the stick. Notably, the local frame $\mathcal{F}$ and keypoint $\mathbf{k}_1$ are located near the contact point, while $\mathbf{k}_2$ is the farthest point on the stick from the paper roll.

Similarly, in Figures 4.20a, 4.21a, 4.23a and 4.24a, local frames are constructed on the master objects (kettle, tissue box, teacup, and rack, respectively) with three p2p constraints extracted to fully constrain the pose of the slave objects (hand, hand, kettle, and hat, respectively).

Task reproduction: Following Section 4.3.3, priorities of the three keypoints are ranked as $\mathrm{Pri}_1 > \mathrm{Pri}_2 > \mathrm{Pri}_3$. This reflects that $\mathbf{k}_1$ is typically the contact point between objects, allowing KAC to reproduce $\mathbf{k}_1$'s motion more accurately than $\mathbf{k}_2$ and $\mathbf{k}_3$.

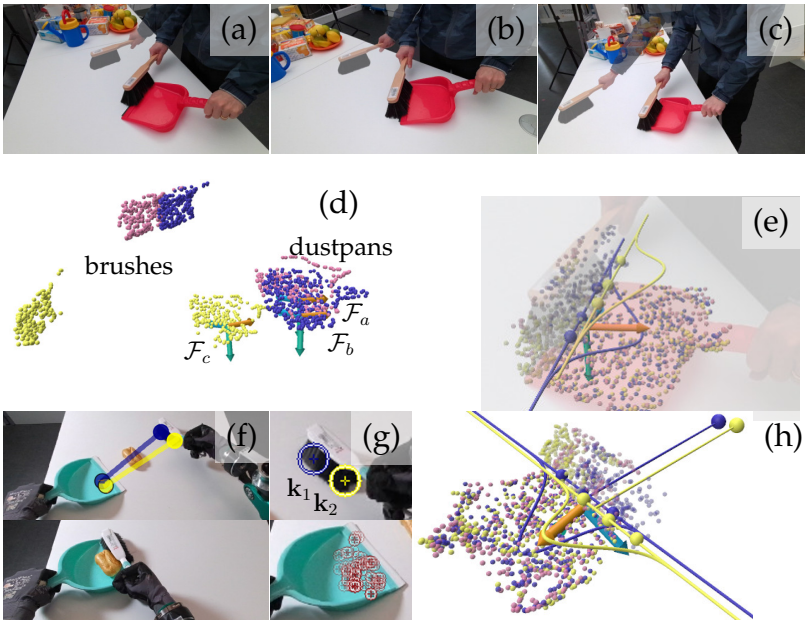
The first five columns of Table 4.1 show examples of reproducing the insertion task learned from a single demonstration (IS ❶), and reproductions without priority in KAC as an ablation study ($\mathsf{IS}_{\mathsf{np}}$ ❶). We display cases with a short stick (#10), a long stick (#6, longer than demonstration sticks), and an extra-long stick (concatenation of #8 and #9), with the largest length difference being approximately $300\,\mathrm{mm}$.

The first row demonstrates Uni-KVIL's ability to correctly adapt task representations of the insertion task to new scenes. This includes estimating the local frame $\mathcal{F}$ on the paper roll, establishing three p2p geometric constraints in $\mathcal{F}$, identifying keypoints on the sticks, and generating VMP trajectories.

Importantly, without priority in KAC, keypoint $\mathbf{k}_1$ in $\mathsf{IS}_{\mathsf{np}}$ ❶-*short* cannot reach its target as accurately as in IS ❶-*short*. Moreover, IS ❶-*long* executes successfully without collision between stick and paper roll due to KAC's priority mechanism, unlike $\mathsf{IS}_{\mathsf{np}}$ ❶-*long*. However, both IS ❶-*ext. long* and $\mathsf{IS}_{\mathsf{np}}$ ❶-*ext. long* result in collisions due to the extreme stick length compared to demonstrations, with more severe collision without priority.

One-shot VIL may generally fail when learned geometric constraints are unreachable, especially with third-person-view demonstrations. For example, task representations learned from single demonstrations of PB and FT (Figures 4.20a and 4.21a) fail during reproduction (tasks A: PB ❶ and D: FT ❶ in Table 4.2) due to unreachable target keypoint positions. This indicates that one-demonstration task representations may not be sufficiently generalizable for motion reproduction.

Summary: Despite some execution failures, Uni-KVIL consistently adapts the tasks to new scenes. Uni-KVIL reliably identifies keypoint positions, locates their targets in the detected local frames, and generates corresponding VMPs. This level of generalization for one-shot VIL is achieved leveraging the combination of proposed task representation with dense visual correspondence models.



Note: Adapted from Gao et al. (2023). © 2023 IEEE.

Figure 4.25.: Uni-KVIL handles viewpoint mismatch in the three demonstrations (a)-(c) by aligning the corresponding local frames on the master object dustpan in (d), which results in an aligned viewpoint in (e). Two p2l constraints and their probability density functions on the principal lines are visualized. The robot reproduces the CT ❸ task from a new viewpoint with a novel brush and dustpan (f), with the keypoints ($\mathbf{k}_1$, $\mathbf{k}_2$) detected on the brush hair (g). The local frame on the dustpan is determined by the $Q = 50$ neighboring points as shown in (g). (f) and (h) depict the keypoints and their movement primitives in 2D and 3D respectively.

Furthermore, the prioritized KAC enables handling shape variations in categorical objects through keypoint target position extrapolation. However, very large shape variations remain challenging. In other words, single demonstrations limit the learning of embodiment-independent generalizable task representations, motivating our evaluation of performance with multiple demonstrations.

Handling viewpoint mismatch

When demonstration videos come from different viewpoints, as in the CT ❸ task (Figure 4.25), we need to align demonstrations into a common viewpoint. As explained in Section 4.2.1, we have in total $|\mathcal{P}_m|$ candidate local frames – i. e., common viewpoints – on the master object. We project the slave objects' motions (e. g., brush) into each candidate local frame $\hat{\mathcal{F}}_j$ (e. g., dustpan) and then apply PCE to extract task representations. PCE jointly extracts keypoints, their

constraints, and the local frame in which these constraints are most prominent. This resolves the viewpoint mismatch problem.

As shown in Figure 4.25e, the geometric constraints become obvious in the aligned viewpoint, which is not the case in Figure 4.25d. Uni-KVIL extracts two p2l constraints for the CT ❸ task, which together approximately form a *parallel constraint*. The estimated probability density functions of the two keypoints on their corresponding principal lines ensure their target positions lie above the dustpan edge.

Notably, using Uni-KVIL's task representation, not only can demonstrations be recorded from different viewpoints, but the robot's reproduction can also be performed from a viewpoint significantly different from any demonstration, as shown in Figures 4.25f to 4.25h. For clarity, we discuss the remaining evaluation results in aligned viewpoints, although demonstrations and reproductions occur in different viewpoints as shown for the CT ❸ task.

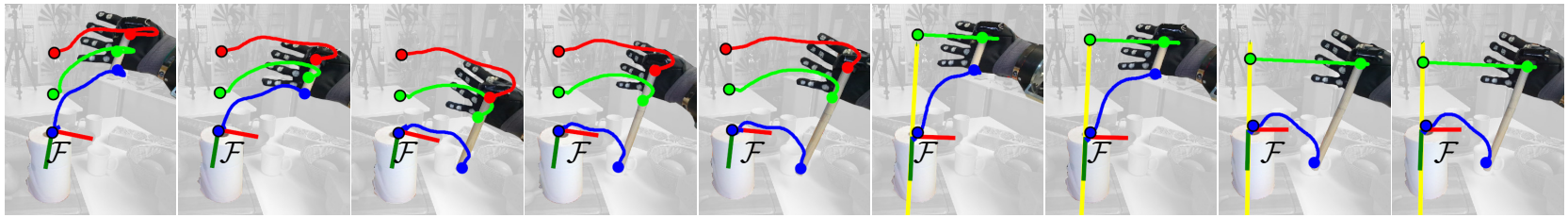
Updating Constraints Incrementally

Here, we apply Uni-KVIL to few-shot imitation learning scenarios where additional demonstrations are incrementally provided for each task. Task constraints in the form of various linear and non-linear principal manifolds are extracted based on variance criteria (see Sections 4.2.1- 4.2.1). Figures 4.20 to 4.24 show the constraints for each task in $\mathcal{T}_E$ extracted by Uni-KVIL from several demonstrations

Task extraction and reproduction of insertion tasks IS. Multiple demonstrations allow consideration of task variations and extraction of prioritized geometric constraints. For example, keypoint $\mathbf{k}_1$ in Figure 4.22b is the most invariant point on the stick across demonstrations, while $\mathbf{k}_2$ is subject to a p2l constraint – contrasting with Figure 4.22a, where all keypoints were subject to p2p constraints.

With this task representation, KAC successfully handles all stick lengths by fulfilling the p2p and p2l constraints, as shown in Table 4.1 for IS ❸ (P). Despite reduced accuracy and precision (see Section 4.4.3, Table 4.7), KAC without priority still achieves successful task completion (see IS_{np} ❸ (Q) in Table 4.1).

Overall, Uni-KVIL's extrapolation abilities significantly improve with three demonstrations versus only one. Due to the nature of the p2l constraint and KAC's priority mechanism, sticks of arbitrary length can be handled. As the line manifold approximately passes through both $\mathbf{k}_1$ and $\mathbf{k}_2$, Uni-KVIL implicitly learns a *collinear constraint* for these keypoints.

Tasks	N: IS ❶	O: IS_{np} ❶	N: IS ❶	N: IS ❶	O: IS_{np} ❶	P: IS ❸	Q: IS_{np} ❸	P: IS ❸	Q: IS_{np} ❸
stick	short	short	long	ext. long	ext. long	short	short	ext. long	ext. long
	3×p2p					p2p, p2l			
TR									
Reproduction									

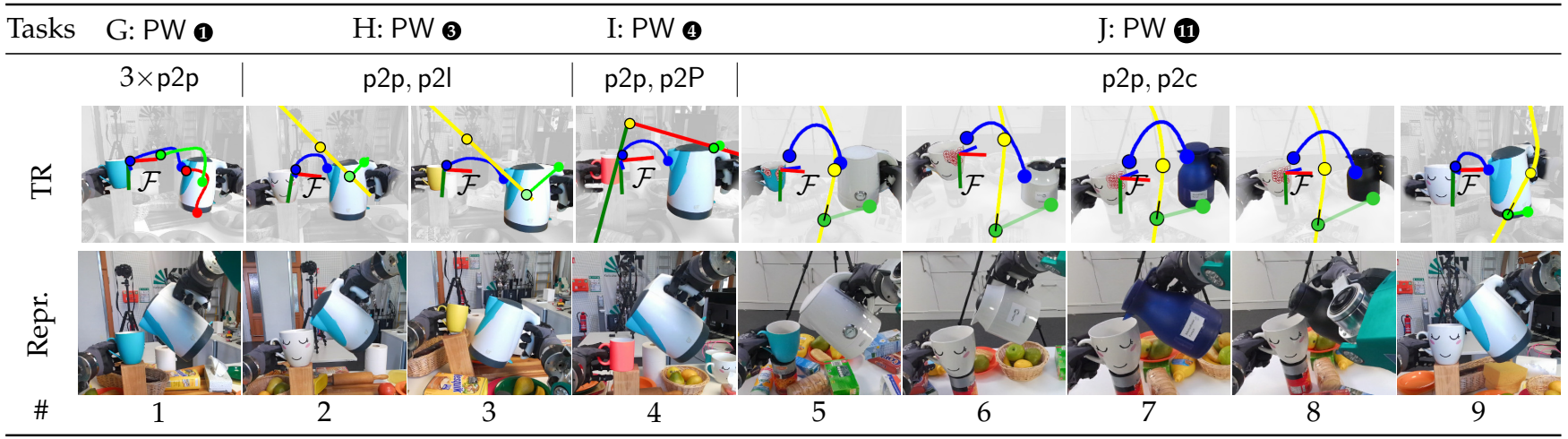
Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Table 4.1.: Reproduction of the insertion tasks with/without priorities. Given an image of the scene before execution, Uni-KVIL's task representation (TR) of each task is used to identify the local frame $\mathcal{F}$, the keypoints ($\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$), their targets positions ($\mathbf{k}_1^g$, $\mathbf{k}_2^g$, $\mathbf{k}_3^g$), their movement primitives (—, —, —), and the line principal manifold (—) in tasks P and Q. The short, long and extremely long sticks correspond to sticks #10, #6, and the concatenation of #8 and #9. Task names and statistics are listed in Table 4.7. The subscript np indicates that the task was reproduced without priority in KAC. The figures in each column are from one of the 20 trials for each task.

Tasks	A: PB ❶	B: PB ❸	C: PB ❹	D: FT ❶	E: FT ❸	F: FT ❹
	3×p2p	p2p, p2l	p2p	3×p2p	p2p, p2l	p2p
TR						
Reproduction						

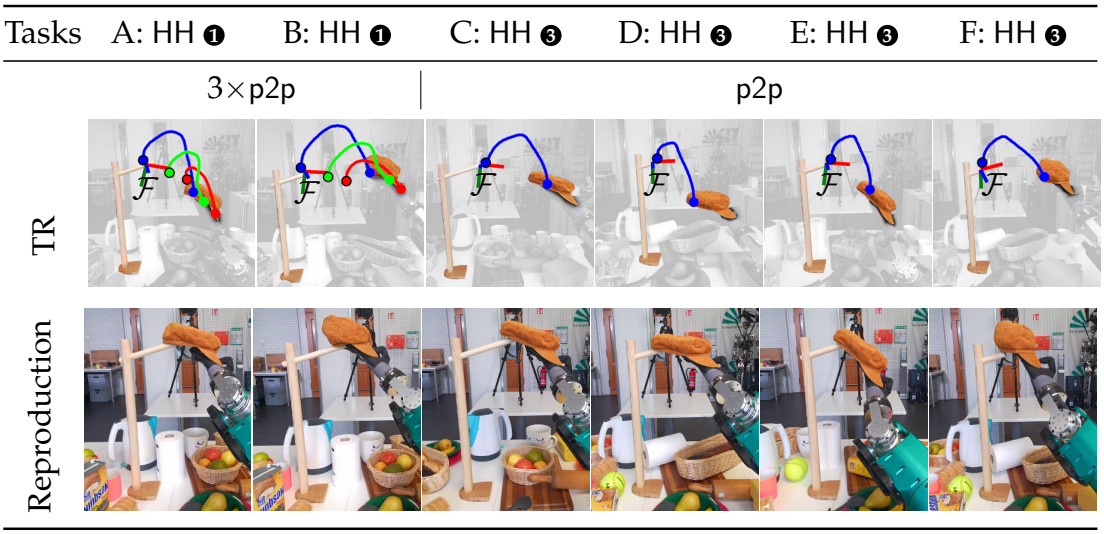
Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Table 4.2.: Reproductions of tasks PB and FT learned from a third-person view without enough demonstrations lead to failure, due to unreachable geometric constraints, see tasks A, B, D, E. Generalizable task representations of PB and FT learned from enough demonstrations can be successfully executed in C and F. Legend as in Table 4.1.



Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Table 4.3.: Reproductions of tasks PW learned from different number of demonstrations. For task H, I and J, we mark the learned principal manifold $\mathcal{M}_{\text{line}}$ (—), $\mathcal{M}_{\text{plane}}$ (—, —) and $\mathcal{M}_{\text{curve}}$ (—), respectively. Additionally, the point (○) indicates the mean of the demonstrated targets of keypoint $\mathbf{k}_2$ on the corresponding principal manifolds, which attracts the keypoint's target point (●) on the manifold. The last row numbers the trails. Other legends as in Table 4.1.



Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Table 4.4.: Reproductions of tasks HH. Legend as in Table 4.1.

Task extraction and reproduction of PB **and** FT **tasks.** For these tasks, three demonstrations are insufficient for complete task representation. In these cases, Uni-KVIL may extract superfluous p2l constraints (see $\mathcal{M}_{\text{line}}$ in Figure 4.20b and Figure 4.21b). This forces the robot to position its hand similarly to the human demonstrator. However, due to third-person demonstrations, the robot cannot achieve this, resulting in failed executions of tasks PB ❸ and FT ❸, similarly to PB ❶ and FT ❶ (see Table 4.2).

These unnecessary p2l constraints are eliminated with an additional demonstration (see Figures 4.20c and 4.21c), enabling successful task reproduction (see PB ❹ and FT ❹ in Table 4.2).

Task extraction and reproduction of PW **tasks.** Similarly to the PB and FT tasks, learning PW task from three demonstrations results in superfluous p2p and p2l constraints. Though the task may still be executable (see the 2$^{\text{nd}}$ trail in Table 4.3), these constraints can lead to collisions between the kettle and environment in other scenarios. For instance, in the the 3$^{\text{rd}}$ trail, with certain initial kettle poses, the generated VMPs and line constraints cause reversed kettle rotation during execution, leading to reproduction failures.

These restrictive representations are mitigated by providing an additional demonstration, which updates problematic constraints to different types – e. g., the p2l constraint in Figure 4.23b becomes a p2P constraint in Figure 4.23c. This significantly improves the pour motion, as the p2P constraint appropriately constrains the kettle in a vertical plane while being less restrictive than the previous p2l constraint. The density force within the plane constraint ensures the kettle's

tilting angle resembles the demonstrations. A successful reproduction of these task representations is shown in Table 4.3-(PW ❹).

With sufficient demonstrations ($N = 11$ in Figure 4.23d), Uni-KVIL extracts a p2c constraint for keypoint $\mathbf{k}_2$ at the kettle's bottom. Reproduction results with different kettle instances are shown in Table 4.3-(PW ⓫). Intuitively, the p2c constraint better aligns with our understanding of pouring. Moreover, the last column shows KAC's ability to correct kettle pose despite a tilted initial position.

The PW tasks in Table 4.3 demonstrate that our approach generalizes well to categorical objects with varying colors, sizes, and shapes, and remains robust to background and viewpoint changes.

Task extraction and reproduction of HH tasks. Unlike the PB, FT, and PW tasks, learning HH task from three demonstrations does not produce superfluous p2l constraints. Instead, due to obvious hat pose variations, Uni-KVIL consistently extracts a single keypoint $\mathbf{k}_1$ on the hat's backside with a p2p constraint represented in the local frame near the hanging stick's end.

As shown in Table 4.4, target poses for the hat in HH ❸ vary according to different initial poses, while in HH ❶, they are fully determined by three p2p constraints. Moreover, as shown in Figures 4.24c to 4.24e, by introducing shape variations of the rack with stick #7 and #8 in Figure 4.24c, the keypoint $\mathbf{k}_1$ only shows position invariances in the extract local frame. This demonstrates that object shape variations in the demonstrations allow for a more precise selection of the local frames that align with our intuition.

Evaluation summary

Our experiments demonstrate that Uni-KVIL's task representations enable successful learning of diverse tasks and their reproduction in new cluttered scenes with significant shape and pose variations in categorical objects. Unlike approaches such as Pari et al. (2022); Yang et al. (2022), our method does not require maintaining the same viewpoint between demonstrations and reproductions, offering greater flexibility. We now discuss how the number and diversity of demonstrations influence these task representations.

Limitations of scarce demonstrations. Task representations learned from a few demonstrations hinder Uni-KVIL's performance and extrapolation ability in three ways: 1) they may be embodiment-dependent and thus unreproducible by the robot, as in Table 4.2 for PB ❶, PB ❸, FT ❶, and FT ❸; 2) they may cause

		Number of demonstrations (N)				
$\mathcal{T}_E$		1	3	4	5	11
a	PB	$3 \times$ p2p	p2p, p2l	p2p	p2p	p2p
b	FT	$3 \times$ p2p	p2p, p2l	p2p	p2p	p2p
c	PW	$3 \times$ p2p	p2p, p2l	p2p, p2P	p2p, p2P	p2p, p2c
d	HH	$3 \times$ p2p	p2p	p2p	p2p	p2p
e	IS	$3 \times$ p2p	p2p, p2l	p2p, p2l	p2p, p2l	p2p, p2l
f	CT	$3 \times$ p2p	p2l, p2l	p2l, p2l	p2l, p2l	p2l, p2l

Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Table 4.5.: Extraction tasks and the geometric constraints of each task learned from different number of demonstrations. We mark the cases ( ) where the learned task representations converge, and highlight the cases (in **blue**) where the learned task representations are generalizable.

collisions during execution, as in IS ❸-*ext. long* (see Table 4.1) and PW ❸ with improper kettle starting pose (see Table 4.3); 3) even when successfully reproduced, they yield reduced control accuracy compared to representations learned from more demonstrations (see Section 4.4.3). These limitations motivate our evaluation of the minimum number of demonstrations required for generalizable task representations according to criteria in Section 4.4.1.

Adequate number of demonstrations. Table 4.5 summarizes extracted geometric constraints for the extraction tasks across different numbers of demonstrations. Notably, learned task representations converge after a certain number of demonstrations: $N = 4$ for PB and FT, $N = 11$ for PW, and $N = 3$ for HH and IS.

Furthermore, representations become generalizable almost simultaneously with convergence. As an exception, PW representations already generalize with p2p and p2P constraints learned from just four demonstrations. Importantly, generalizable representations do not necessarily require p2p constraints, as evidenced by the CT task represented by two p2l constraints.

Overall, our approach efficiently extracts generalizable task representations from considerably fewer demonstrations than state-of-the-art methods.

Object pose and shape variations. Uni-KVIL's task representations fundamentally rely on variations observed in demonstrations to extract appropriate constraints. For example, in PB, the demonstrator's hand pose variations (Figure 4.20c) allow Uni-KVIL to distinguish the middle finger's tip (used for button pressing) from other candidate points on the hand. These variations enable efficient extraction of keypoint $\mathbf{k}_1$ subject to a p2p constraint.

index	$\mathcal{T}_E$	role	object	PV	SV	TR
1	PB	master	kettle	-	✗	Figures 4.20b and 4.20c
		slave	hand	✓	✗	
2	FT	master	tissue box	-	✓	Figures 4.21b and 4.21c
		slave	hand	✓	✗	
3	PW	master	teacups	-	✓	Figures 4.23b to 4.23d
		slave	kettle	✓	✓	
4	HH	master	rack	-	✗	Figure 4.24b
		slave	hat	✓	✗	
5	HH	master	rack	-	✓	Figure 4.24c
		slave	hat	✓	✗	
6	HH	master	rack	-	✗	Figures 4.24d and 4.24e
		slave	hat	✓	✓	
7	IS	master	paper roll	-	✗	Figures 4.22b and 4.22c
		slave	stick	✗	✓	
8	CT	master	dustpan	-	✓	Figure 4.25
		slave	brush	✓	✓	

Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Table 4.6.: Pose variations (PV) and shape variations (SV) in the demonstrations along with the detected master and slave objects. The corresponding task representations (TR) are linked in the last column. Pose variations of the master objects are not relevant (-) as the local frames representing the object pose are constructed on the masters.

Pose variations also provide the spatial distribution of keypoints across demonstrations. This allows Uni-KVIL to associate keypoint $\mathbf{k}_2$ with geometric constraints such as p2l (Figures 4.23b and 4.25e), p2P (Figure 4.23c), and p2c (Figure 4.23d).

Shape variations likewise facilitate efficient extraction of keypoints and geometric constraints. For instance, stick length variations in the IS task enable Uni-KVIL to extract keypoints $\mathbf{k}_1$ and $\mathbf{k}_2$ subject to p2p and p2l constraints, respectively, as shown in Figure 4.22b.

In other cases, shape variations in master objects help eliminate redundancy in candidate local frames. For example, all 300 candidate local frames on the rack in HH (Figure 4.24b) are equivalent due to the absence of rack variations across demonstrations. Here, we empirically selects local frame $\mathcal{F}$ as the closest on average to keypoint $\mathbf{k}_1$ on the hat.

This redundancy is resolved by introducing shape variations in master objects. For example, considering two rack variations in HH (Figure 4.24c), $\mathbf{k}_1$ is position-invariant only in local frames near the contact point between rack and

hat. This allows Uni-KVIL to focus on these frames and filter out others. Figure 4.24d shows that the representation converges and remains consistent with Figures 4.24b and 4.24c even with more demonstrations containing significant hat shape and pose variations.

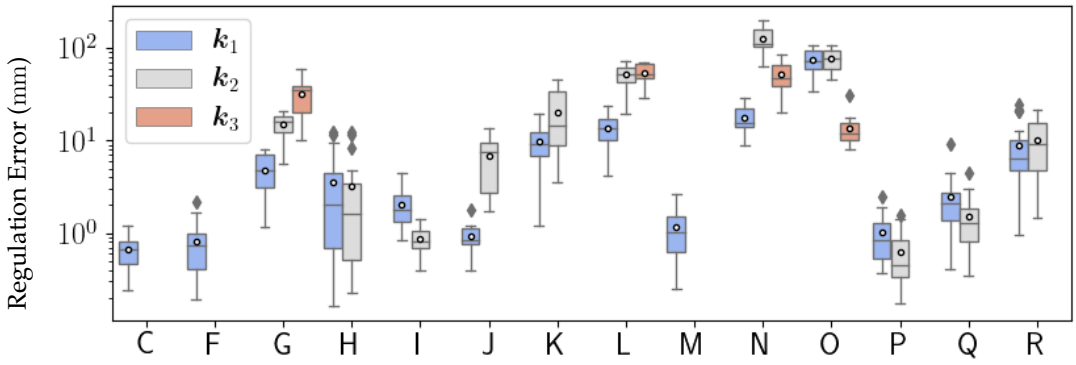
Similarly, tissue box shape variations in FT enable selection of local frames around the grasping point, while teacup shape variations in PW ensure local frames for pouring are positioned around the rim.

Table 4.6 summarizes the impact of pose and shape variations on the considered tasks. Uni-KVIL effectively handles and leverages pose and shape variations in master and slave objects during demonstrations, enabling robust extraction of local frames, keypoints, and geometric constraints. These results highlight that beyond generalizable task representations, the quality of human demonstrations plays a critical role for data-efficient Visual Imitation Learning (VIL) approaches.

4.4.3. Evaluation of KAC

In Section 4.4.2, we demonstrated Uni-KVIL's ability to extract generalizable task representations from a few demonstrations. Specifically, we showed these task representations successfully adapt to new cluttered scenes with categorical objects. To reliably execute these learned tasks, we propose the prioritized keypoint-based admittance controller (KAC) in Section 4.3, which effectively handles varying numbers of keypoints across different task representations, as shown in Table 4.5. In this section, we evaluate the proposed KAC in terms of control accuracy, precision (i. e., repeatability), and success rate across the 18 tasks described in Section 4.4.1 (see also Table 4.7).

Following the evaluation protocol in Section 4.4.1, each task is reproduced $N_r = 20$ times. We record the trajectories of all relevant keypoints during execution. Since we are particularly interested in KAC's regulation behavior when keypoints satisfy their corresponding geometric constraints (i. e., when the VMPs finish), we record keypoint trajectories for an additional $2\,\mathrm{s}$ time window, denoted by $\mathscr{T}_{\text{end}}$. These trajectories are then compared to their corresponding target trajectories – VMP trajectories for keypoints subject to p2p constraints, and attractor trajectories for keypoints subject to other constraint types. Notably, for the latter, the 3D position of the attractor is recovered from the corresponding 1-dimensional VMP in the orthogonal direction of the corresponding principal manifold (see also Section 4.3.1).



Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Figure 4.26.: Regulation errors between each keypoint and its target estimated in $\mathscr{T}_{\text{end}}$ over $N_r = 20$ trials for tasks C to R in Table 4.7. The mean regulation error, i.e., control accuracy Acc., is depicted as $\circ$. The box shows the first and third quartiles of the regulation error of each keypoint, with the bar inside it indicating the median.

First, we evaluate KAC's ability to satisfy the learned keypoints' geometric constraints by computing the regulation error of each keypoint during $\mathscr{T}_{\text{end}}$ for each trial:

$$e_v = \frac{1}{T_r} \sum_{t=0}^{T_r} \|\mathbf{k}^g_{l,v}(t) - \mathbf{k}_{l,v}(t)\|_2, \quad v \in [1, N_r] \cap \mathbb{Z},$$

where T_r represents the total timesteps recorded in $\mathscr{T}_{\text{end}}$, and $\mathbf{k}_{l,v}, \mathbf{k}^g_{l,v}$ are the recorded and target positions of the considered keypoint in the v^{th} trial, respectively. Figure 4.26 displays the distribution of keypoint regulation errors across 20 trials for each task, with mean values ($\circ$) corresponding to control accuracy:

$$\mathsf{Acc} = \frac{1}{N_r} \sum_{v=0}^{N_r} e_v.$$

Second, we evaluate KAC's control precision (i. e., repeatability) for all keypoints across all tasks, computed as:

$$\mathsf{Prec} = \sqrt{\frac{1}{N_r} \frac{1}{T_r} \sum_{v=1}^{N_r} \sum_{t=1}^{T_r} \|\mathbf{k}_{l,v}(t) - \boldsymbol{\mu}_{\mathbf{k}_{l,v}}\|_2^2}, \tag{4.37}$$

where $\boldsymbol{\mu}_{\mathbf{k}_{l,v}} = \frac{1}{T_r} \sum_{t=1}^{T_r} \mathbf{k}_{l,v}(t)$. Finally, we report the success rate for each task according to the evaluation protocols described in Section 4.4.1.

Table 4.7 presents the evaluation results of KAC across all three metrics. As indicated by qualitative evaluations in Table 4.2, the tasks PB and FT result

in a 0% success rate when learned from fewer than four demonstrations (see A-B, D-E in Table 4.7). As discussed in Section 4.4.2, this failure stems from geometric constraints that are unreachable for the robot. In contrast, task representations learned from four demonstrations are generalizable (see Table 4.5 in Section 4.4.2), enabling KAC to achieve sub-millimeter control accuracy and precision, as well as $\geq 90\%$ success rates (see C and F in Table 4.7).

As evidenced in Figure 4.26 and Table 4.7 (G-J), PW ❹ and PW ⓫ outperform PW ❸ in terms of control accuracy, precision, and success rate. This superiority stems from the fact that, unlike PW ❸, both PW ❹ and PW ⓫ are generalizable (see Table 4.5 and Table 4.3). Although PW ❶ exhibits relatively high control precision and success rate, its control accuracy remains low, and the kettle's pose is fully constrained by three p2p constraints, which significantly limits its extrapolation capabilities. Interestingly, while $\mathbf{k}_1$'s control accuracy in the PW task increases with the number of demonstrations, $\mathbf{k}_2$'s highest control accuracy is achieved in PW ❹ (I). This results from $\mathbf{k}_2$'s p2P constraint in PW ❹, which is easier to satisfy than the p2c constraint in PW ⓫. For the same reason, we observe lower precision in PW ❹ compared to PW ⓫.

Similar to the pattern observed in PW, HH ❸ (M) and IS ❸ (P) demonstrate higher control accuracy and precision than HH ❶ (L) and IS ❶ (N), respectively. Furthermore, thanks to the priority mechanism introduced in KAC, HH ❶ (L) and IS ❶ (N) achieve 95% and 62% success rates, respectively, despite the availability of only a single demonstration. The lower performance of IS ❶ is attributed to the extremely long stick used in reproduction (see Table 4.1) due to its large shape variation, which demands high generalization capabilities. In contrast, the hat in HH is only slightly deformable with minimal shape variations.

It is important to note that for HH and IS, we do not compare control accuracy between keypoints subject to different geometric constraints (shown in gray in Table 4.7). Since these keypoints are controlled according to different priorities within KAC, their reported accuracy heavily depends on the object shape variations occurring in these tasks. For example, when using sticks of various lengths in the insertion task IS ❶ (N), KAC assigns highest priority to keypoint $\mathbf{k}_1$, ensuring that $\mathbf{k}_1$ is controlled with significantly higher accuracy than $\mathbf{k}_2$ and $\mathbf{k}_3$. Notably, $\mathbf{k}_2$'s highest regulation error in this task is approximately $150\,\mathrm{mm}$, which corresponds precisely to the maximum length difference between sticks used in demonstration versus reproduction. Therefore, in HH and IS, the control accuracy of $\mathbf{k}_2$ and $\mathbf{k}_3$ is more influenced by experimental setups (e. g., stick lengths) than by KAC itself. In CT ❸, where two p2l constraints share the same priority, both keypoints are equally controlled toward regions with high likeli-

			Acc. (mm)			Prec. (mm)			
	$\mathcal{T}_R$	TR	$\mathbf{k}_1$	$\mathbf{k}_2$	$\mathbf{k}_3$	$\mathbf{k}_1$	$\mathbf{k}_2$	$\mathbf{k}_3$	R (%)
A	PB ❶	Figure 4.20a	×	×	×	×	×	×	0
B	PB ❸	Figure 4.20b	×	×	×	×	×	×	0
C	PB ❹	Figure 4.20c	0.67	-	-	0.34	-	-	90
D	FT ❶	Figure 4.21a	×	×	×	×	×	×	0
E	FT ❸	Figure 4.21b	×	×	×	×	×	×	0
F	FT ❹	Figure 4.21c	0.82	-	-	0.65	-	-	100
G	PW ❶	Figure 4.23a	4.81	15.01	32.09	0.75	0.78	0.72	95
H	PW ❸	Figure 4.23b	3.56	3.21	-	1.20	2.12	-	87
I	PW ❹	Figure 4.23c	2.01	0.88	-	1.04	5.67	-	94
J	PW ⓫	Figure 4.23d	0.93	6.80	-	0.42	0.56	-	100
K	PW_{np} ⓫	Figure 4.23d	9.71	20.08	-	0.90	1.16	-	100
L	HH ❶	Figure 4.24a	13.53	51.33	54.28	0.67	0.67	0.67	95
M	HH ❸	Figure 4.24b	1.48	-	-	0.47	-	-	95
N	IS ❶	Figure 4.22a	17.77	125.40	51.98	1.55	1.30	1.34	62
O	IS_{np} ❶	Figure 4.22a	74.55	77.05	13.40	1.03	0.97	1.02	25
P	IS ❸	Figure 4.22b	1.01	0.63	-	0.36	0.36	-	95
Q	IS_{np} ❸	Figure 4.22b	2.46	1.50	-	2.32	1.39	-	90
R	CT ❸	Figure 4.25e	8.89	10.22	-	2.04	2.03	-	95

Note: Reprinted from Gao et al. (2023). © 2023 IEEE.

Table 4.7.: Evaluation of KAC in terms of control accuracy (Acc.), precision (Prec.), and success rate (R) for each category of tasks with task representations (TR) learned from different numbers Ⓝ of demonstrations. Ablation studies (denoted by $\cdot_{\text{np}}$) are also conducted in tasks K, O, and Q by removing the priority (see Section 4.3.3) from KAC. The cases where data is not available due to failure execution and where a specific keypoint is not required are denoted as × and -, respectively. Gray numbers in N and O indicate inconsequential values.

hood on their corresponding principal lines. The two spring-damper systems for these keypoints reach equilibrium, resulting in comparable control accuracy and precision.

Compared to IS ❶, the ablation study IS_{np} ❶ (O), conducted without KAC's priority mechanism, shows a significant drop in success rate (down to 25%), as $\mathbf{k}_3$ achieves the highest accuracy at the expense of $\mathbf{k}_1$ and $\mathbf{k}_2$. This outcome is expected, as the three identical virtual spring-damper systems for $\mathbf{k}_1$, $\mathbf{k}_2$, and $\mathbf{k}_3$ reach equilibrium, with $\mathbf{k}_3$ typically positioned between $\mathbf{k}_1$ and $\mathbf{k}_2$. For the ablation tasks PW_{np} ⓫ (K) and IS_{np} ❸ (Q), which are reproduced from generalizable task representations, removing KAC's priorities does not affect their success

rates. However, both control accuracy and precision decrease compared to their prioritized counterparts PW ⓫ (J) and IS ❸ (P).

4.5. Conclusion and Discussion

In this chapter, we introduced the novel keypoints-based visual imitation learning focusing on unimanual manipulation tasks, named Uni-KVIL. We first propose a *neural-descriptor-based generalizable object representation*, allowing us to detect dense correspondences and transfer keypoints and poses between categorical objects via *object canonical spaces*. This facilitates keypoint-based task representation at finest granularity and intra-category generalization, as well as a robust percpetion pipeline preprocessing human demonstration videos under viewpoint mismatch and occlusion challenges.

Given the high-quality point-based dataset extracted from a small set of human demonstration videos, Uni-KVIL learns *sparse, object-centric, viewpoint invariant*, and *embodiment-independent* task representations. The proposed task representations are based on the extraction of a wide range of *geometric constraints on principal manifolds*, leveraging the proposed *Principal Constraint Estimation* (PCE) algorithm – a data-efficient statistical method that effectively identify invariant task features. PCE jointly extracts *keypoints*, their *constraints* and the *local frames* anchored to master object's functional parts, making the representation object-centric, significantly enhancing the generalization capabilities. Uni-KVIL enables *one-shot* and *few-shot* VIL and incrementally updates learned task representations when additional demonstrations are provided, thereby enhancing their extrapolation capabilities. Uni-KVIL's task representations also incorporate task-specific keypoint control policies encoded as VMPs, which are leveraged for task execution through a *prioritized keypoint-based admittance controller* (KAC). Compared to control policies based on RL or visual servoing, our approach allow flexible temporal scaling and support via-point adaptation (including start and target positions). Consequently, they substantially contribute to Uni-KVIL's generalization capabilities by extrapolating keypoint target positions on the learned principal manifold.

Our evaluation demonstrates that Uni-KVIL consistently learned generalizable task representations for six manipulation tasks involving cluttered scenes, new instances of categorical objects, and significant variations in object poses and shapes. Importantly, we showed that the learned task representations converge

and become generalizable with substantially fewer demonstrations than state-of-the-art approaches such as Sermanet et al. (2018); Sharma et al. (2019); Pathak et al. (2018); Jin and Jagersand (2022). Interestingly, the sparse keypoint-based geometric constraints extracted by Uni-KVIL largely aligned with human intuition. These include the extraction of a single p2p constraint for button pressing, a pair of p2p and p2l constraints for the insertion task, and a p2p coupled with a p2c constraint for the pouring task, among others.

It is important to emphasize that the decomposed control and priority mechanism of KAC significantly enhanced Uni-KVIL's extrapolation capabilities. Our quantitative evaluations demonstrated its ability to reproduce learned task representations with high control accuracy, precision, and success rates. Notably, Uni-KVIL accurately handled extensive shape variations in the insertion task. In contrast, previous works either did not address or only briefly discussed the extrapolation capabilities of their approaches (Sieb et al., 2019; Yang et al., 2022; Jin and Jagersand, 2022; Karnan et al., 2022a). For instance, Jin & Jagersand Jin and Jagersand (2022) only demonstrated extrapolation to another instance of the hammer category with minimal shape variation, without providing any quantitative evaluations.

It is essential to recognize that variations in object poses and shapes play a critical role in learning generalizable task representations, especially when only a limited number of demonstrations are available. Without such variations, our approach can still generalize to categorical objects through dense neural descriptors but achieves lower control accuracy, precision, and success rates, and may fail in extreme cases, such as in one-shot VIL scenarios.

Uni-KVIL faces limitations in terms of bimanual manipulation, learning from demonstrations containing a series of actions, neural descriptor models, and inter-category generalization.

Bimanuality: In this chapter, we focused on unimanual manipulation tasks involving at most two objects, modeled within a single-layer master-slave relationship. While our generalizable object representation enables the hierarchical decomposition of scenes involving multiple objects, this capability has not yet been fully exploited.

In Chapter 5, we extend Uni-KVIL to bimanual manipulation tasks by introducing a hierarchical structure of master-slave object relationships. This allows us to model various bimanual coordination strategies (Krebs and Asfour, 2022). Furthermore, bimanual manipulation often demands precise spatial coordination, which is essential for tasks with various execution styles. For instance, pouring

beer, requires tilting the glass and aligning the beer with the glass's side, which differs from the pouring water tasks evaluated in this chapter. Therefor, we explore fine-grained task execution styles in this context.

Motion segmentation: Currently, the motion segments of interest are manually extracted from the demonstration videos. For example, in the pouring task, we only consider the segment where the kettle is moved to the cup and the pouring action is performed, excluding the reaching and placement segments. To automate the extraction of these motion segments and align semantically identical segments across multiple demonstrations for learning Uni-KVIL's task representation, a motion segmentation algorithm is necessary. As we also focus on bimanual manipulation in Chapter 5, this segmentation algorithm should robustly identify segments to facilitate the extraction of bimanual spatio-temporal coordination strategies. To address this, we introduce a motion segmentation algorithm in Chapter 6.

Neural Descriptor Models: We proposed a generalizable object representation leveraging both 2D and 3D neural descriptors and object canonical space. However, these descriptors are used separately, each with its own limitations. For instance, 2D descriptors require keypoints to be located on object surfaces, limiting Uni-KVIL's ability to find occluded keypoints. On the other hand, MIMO-based 3D descriptors necessitate extensive training on known object categories, hindering application to open-vocabulary settings.

Relying on multiple descriptor models not only requires extensive GPU memory but also slows down processing speed, which hinders online visual imitation learning. A lightweight neural descriptor model encompassing both 2D and 3D features with competitive performance has not yet been thoroughly explored in the literature.

Looking forward, we believe that dense correspondence models should account for multiple data modalities, improve granularity, and enhance viewpoint consistency for more reliable knowledge transfer. It is important to note that the visual descriptor models are interchangeable, and more informative models can be explored in future work.

Inter-category generalization: A key challenge in (visual) imitation learning is bridging the gap between symbolic and subsymbolic task representations. Keypoints, which encode subsymbolic parameters, are essential for task execution but often lack a clear semantic interpretation. This limitation hinders the learning of comprehensive task models. For instance, in a pouring task, the contain affordance implies that the kettle's spout – corresponding to the pour affordance

– should be positioned above the contain affordance region (Hadjivelichkov et al., 2022). However, semantic representations alone cannot fully describe different pouring styles like the pouring beer example above. To accommodate such variations, additional subsymbolic parameters must be incorporated.

Uni-KVIL addresses the subsymbolic aspects of task modeling by extracting keypoint-based geometric constraints and generalize them to categorical objects via visual correspondences. To further enhance Uni-KVIL's generalization capability, an extraction method (Jiang et al., 2021) could be incorporated to establish links between extracted keypoints and symbolic task representations. For example, keypoint neighborhoods in the semantic descriptor space can suggest affordance regions (Hadjivelichkov et al., 2022; Do et al., 2018). Understanding the concept of affordance allow transferring a learned pouring water task to similar tasks like pouring milk into a bowl. Additionally, spatial relations are commonly used to model the distribution of affordance regions or object instances (Kartmann et al., 2021), which facilitate transferring of tasks such as from "placing an apple on the right of a bowl" to "placing a banana on the left of a cup". These semantic concepts improve the generalization of learned task representations across different object categories and tasks. We will consider these extensions as future work.

CHAPTER 5

Learning Bimanual Coordination Strategies

Bimanual manipulation is essential to human daily activities, allowing tasks that require precise spatio-temporal coordination of both arms, such as cooking, assembling, and transporting. These tasks often involve complementary or synchronized movements, with each limb potentially taking on a distinct role depending on the task.

Human activities rarely exist as purely unimanual or bimanual actions. Instead, most complete task executions involve a fluid combination of both modalities. Moreover, certain tasks can be performed either unimanually or bimanually, depending on the context. For instance, pouring milk into a bowl typically requires unimanual skill, while pouring beer often engages both arms – one arm stabilizing the tilted cup while the other performs the pouring action. As shown in Figure 1.1, the pouring water action can be performed both with a single arm or with both arms depending on the context. This flexibility demonstrates the adaptability and fine-grained motion details inherent in human manipulation skills.

As highlighted by Krebs and Asfour (2022), bimanual coordination can be categorized into distinct categories: uncoordinated, loosely coupled, and tightly coupled coordination. This categorization underscores that bimanual manipulation is not simply the sum of two unimanual task representations, but rather a complex interplay of coordinated actions.

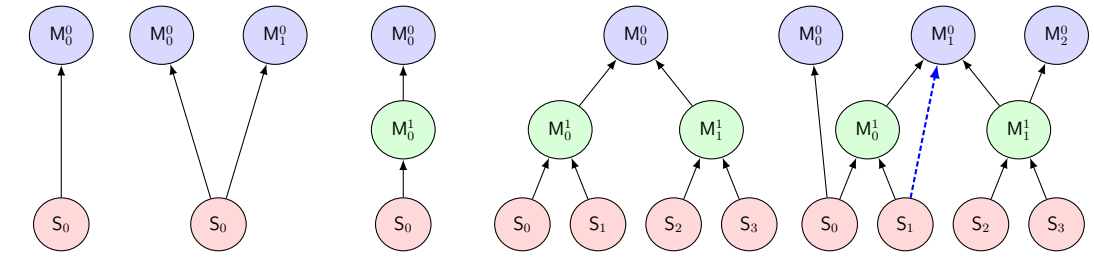
The unique challenges of bimanual manipulation stem from its inherent complexity involving intricate object relationships, fine-grained motion details, and diverse coordination strategies between arms. Visual imitation learning provides a powerful approach to teaching robots new skills from human demonstrations, leveraging visual data to capture the complex spatio-temporal relationships intrinsic to bimanual tasks. This approach not only reduces the need for explicit programming of coordination strategies and control policies but also facilitates the generalization of learned skills to novel scenarios. However, integrating visual perception with bimanual coordination and control introduces unique challenges, such as modeling coordination and geometric constraints between objects from high-dimensional visual inputs.

Similar to unimanual tasks, bimanual manipulation tasks are typically characterized by invariant task features across multiple demonstrations (Muhlig et al., 2009b,a). For instance, pouring tasks are defined by the alignment of the kettle's spout to the cup's rim, while each hand grasps and controls one object. Although bimanual tasks often involve simultaneous manipulation of two or more objects, this invariant feature allows us to decompose the task to focus on geometric constraints between object pairs. This decomposition yields a graph of object relationships, the structure of which implies different coordination categories.

In this chapter, we propose Bi-KVIL, an extension of Uni-KVIL for keypoint-based learning of bimanual manipulation task representations, which capture all relevant geometric constraints between hands and objects. To address the challenges outlined above, we first introduce the novel hybrid *Master-Slave Relationship* (MSR) in Section 5.1, extending the concept of single master-slave pair of Uni-KVIL (see Section 4.2) to a master-slave graph that encompasses all task-relevant objects. In this extended graph, each master object can also function as a slave object paired with another master object. Consequently, we unify the representation of *roles*, *relationships*, and *task constraints* for both objects and hands. The coordination categories (Krebs and Asfour, 2022) of demonstrated bimanual tasks are then derived from the extracted MSR described in Section 5.2. Finally, we extend the keypoint-based admittance controller of Uni-KVIL to bimanual tasks (Section 5.3), followed by evaluation with various manipulation tasks in Section 5.4. Overall, Bi-KVIL unifies the learning of object-centric uni- and bimanual manipulation tasks and captures fine-grained manipulation styles. Note that we specifically focus on the spatial aspect of bimanual coordination in this chapter, while temporal coordination is addressed in Chapter 6. The contributions of this chapter are summarized in Section 1.2.2. Parts of the content presented in this chapter were previously published by Gao et al. (2024).

5.1. Hybrid Master-Slave Relationship

To systematically analyze object-object and object-hand relationships in manipulation tasks, we propose *five* types of Master-Slave Relationships (MSRs) commonly observed in unimanual and bimanual manipulation: *single*, *multiple*, *multi-level*, *hierarchical*, and *hybrid* MSR (see Figure 5.1). This nomenclature draws inspiration from the definition and graph representation of inheritance in C++ programming language.



(a) Single (b) Multiple (c) Multi-level (d) Hierarchical (e) Hybrid

Note: Adapted from Gao et al. (2024). © 2024 IEEE.

Figure 5.1.: The proposed MSR diagrams. M_j^i represents the j-th master object at level i. For level $i > 0$, a master object is itself a slave object paired to another master object at level $i - 1$. The slave objects are located at the lowest level. The doted blue arrow (⋯›) is only for the proposed MSR, and does not exist in the topology of inheritance in the C++ programming language.

1. A *single* MSR (Figure 5.1a) corresponds the scenario in Uni-KVIL, where only two objects interact within a single master-slave pair. Examples include any single-hand grasping task or a pouring action from a kettle into a cup when considering only the kettle-cup pair.
2. In contrast, *multiple* MSR (Figure 5.1b) describes cases where the motion of a slave object is defined in local frames of multiple master objects. This is particularly valuable in assembly, stacking, or multi-object rearrangement tasks, where the pose of one object depends on multiple other objects.
3. The *multi-level* MSR (Figure 5.1c) extends this concept further by allowing a slave object to simultaneously act as a master for another slave object. For example, when using a broom to sweep debris into a dustpan, the broom acts as both a master object for the hand and a slave object relative to the dustpan.

We can further combine these three building blocks (single, multiple and multi-level) to create more complex object relationships.

4. A *hierarchical* MSR (Figure 5.1d) forms a tree-like structure where each master object can be associated with multiple slave objects.
5. The *hybrid* MSR (HMSR, Figure 5.1e) combines multiple and hierarchical MSRs, resulting in a *Directed Acyclic Graph* (DAG). In this graph, a slave object can have master objects at different levels (e. g., S_1 in Figure 5.1e).

Unlike inheritance in C++, a slave object in the HMSR does not inherit geometric constraints from its master. Instead, each master provides a reference frame only for its direct slave object, with constraints explicitly defined for each object pair (as illustrated by the ⇢ line in Figure 5.1e). For example, in the pouring task (Figure 1.1), the cup is designated as the master object that provides local frames and constraints for the kettle. In turn, the kettle acts as a master object, where its handle provides a local frame to define a grasp constraint for the right hand. Here, the local frames and constraints provided by the cup apply only to the kettle and are not transmitted to the hand via the kettle. More examples are discussed in Section 5.4.1. In the remainder of this chapter, we employ HMSR as a general framework that subsumes all other MSR types.

To extract the HMSR, we initially construct an initial DAG using motion saliency, grasping, and pose invariance detection (see Sections 5.1.1, 5.1.3 and 5.1.4). Each valid master-slave pair in the graph must exhibit at least one of the constraints described in Section 4.2. In Section 5.1.5, we then apply Uni-KVIL on each object pair within the HMSR, truncating those pairs without constraints to obtain a sparse and compact graph.

5.1.1. Absolute Motion Saliency Detection

Any object motion must be represented in a local frame attached to an object in the scene. This reference object can be selected from one of three categories: 1) a static object (e. g., a table), 2) a moving object (e. g., goods on a conveyor belt, dishes on a rotating table, or an object being transferred during handover), or 3) the initial state of an object that is about to be moved. An illustrative example of the third case is lifting a cup from a table and then wiping the area where the cup was initially placed (see Figure 5.2). In this scenario, using the table as the reference object is generally unsuitable, as the wiping location depends on the cup's original position.

We assume that when constraints are extracted between moving and globally static objects, the static object serves as a top-level master object. To distinguish between *static* and *moving* objects, we analyze their point velocities in the global

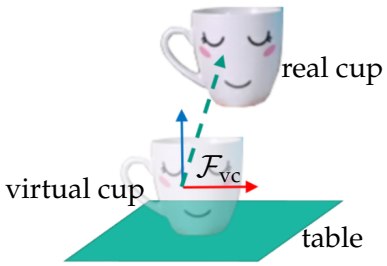


Figure 5.2.: Virtual object: the virtual cup is defined as the initial state of the real cup. The local frame $\mathcal{F}_{vc}$ establishes a reference for representing the motion of the real cup.

coordinate system. In our experiments, the camera frame is used as the global frame since it remains stationary during each recording, though the viewpoint may change across different demonstrations.

Given the 3D velocity trajectories of all candidate points on objects and human hands (see Section 4.1.4), we compute the average norm of the velocity of all points on an object O_i across all timesteps. An object is considered static if this average is below the threshold ξ_{static}, i. e.,

$$\mathcal{O}_{\text{static}} = \left\{ O_i \left| \frac{\sum_{p=0}^{P_i} \sum_{t=0}^{T} \|\dot{\mathbf{p}}_{i,p}(t)\|}{T P_i} < \xi_{\text{static}}, i \in [1, I] \cap \mathbb{Z}, I = |\mathcal{O}| \right. \right\}, \tag{5.1}$$

where T is time horizon, P_i is the number of candidate points on object O_i, and $\dot{\mathbf{p}}_{i,p}(t)$ denotes the velocity vector of the p^{th} candidate point on O_i. All objects not meeting this criterion are classified as moving, i. e., $\mathcal{O}_{\text{moving}} = \mathcal{O} \setminus \mathcal{O}_{\text{static}}$.

5.1.2. Virtual Object

To avoid cases where the table might be inappropriately chosen as the master object (as in the wiping example), we define local frames on the *initial state* of moving objects, with respect to which the constraints and subsequent motion are modeled (see Figure 5.2). Specifically, the *initial state* of each moving object O_i – the sampled set of candidate points on it at the first timestep ($t = 0$) – is treated as a special entity called a *virtual object*, denoted O_i^v. The virtual object retains its initial position throughout the demonstration, thereby keeping its candidate points static.

Intuitively, a virtual object can considered an imaginary object preserving the initial state of the real moving object throughout the demonstration. The virtual

object is treated equivalently to real objects in the HMSR framework; it will be pruned if no clear geometric constraints to other objects are identified.

5.1.3. Grasp Detection

Before analyzing bimanual coordination, it is essential to determine the grasping relationships between hands and objects. Analogous to Uni-KVIL, the human hand is treated as a special object with 21 keypoints (Romero et al., 2017). A firm grasp is detected when the following conditions are met: 1) the hand is in contact with the object, 2) the hand remains relatively static to the object, and 3) a subset of the candidate points on the object is enclosed by the fingers.

Contact between two objects is determined by assessing their relative spatial distances and velocities. Specifically, a hand-object pair is considered in contact when the mean distance between the top-k closest point pairs drops below a threshold and when the average change in these distances (e. g., the velocity) is also below a threshold (see Appendix B.2 for details).

Since changes in distance alone do not capture the complete motion (e. g., a hand sliding on a table or rotating around a button while maintaining contact), we further analyze the relative static status between the object and the hand. For each hand-object pair, the hand candidate points' trajectories are projected onto all candidate local frames on the object, and the velocity trajectories are computed using a first-order Savitzky-Golay filter. The mean norm of the projected hand keypoint velocities in the time-varying top-k closest pairs is then computed (see Appendix B.3 for more details. If this mean velocity is below the relative velocity threshold $\xi_{\text{r,vel}}$, the hand and object are considered to be relatively static.

Finally, we determine whether the convex hull of the finger skeleton encloses any part of the object. If all three conditions (contact, relative static status, and enclosure) are satisfied, a *firm grasp* is detected. In our object-centric representation, grasps are modeled with respect to the grasped object. More specifically, the *grasp constraint* is formulated as *a pose distribution within a local frame anchored to the object's functional parts* (e. g., a handle), analogous to how a slave object's constraints are modeled relative to a master object. Thus, the hand is designated as a slave object paired with the grasped master object.

5.1.4. Pose Invariance Detection

The bottom-up approach of KVIL determines object roles solely from the motion characteristics of sampled points, without any prior semantic knowledge

of object function or task roles. In Uni-KVIL, the master-slave relationship is established based on relative motion saliency (see Section 4.1.4). While this approach is feasible for scenarios involving a single master-slave pair, it does not generalize to bimanual tasks where both objects may move. This raises the question: *Which motion characteristics can reliably determine the master-slave roles in bimanual manipulation?*

Since the motion of object A relative to object B is equivalent to that of B relative to A, a bidirectional MSR might be erroneously constructed, resulting in improper reproductions. For example, in a pouring task, merely tilting a moving cup toward the kettle spout could generate the same relative motion regardless of which object is considered the master, leading to an invalid action.

To resolve this ambiguity, Krebs and Asfour (2022) proposed designating the master object as the less mobile one, based on absolute motion saliency. However, this assumption may fail without clearly defining the period of the task during which motion saliency is measured. In practice, the master object may sometimes move faster or farther than the slave – for instance, a cup is typically considered the master in pouring tasks even if it travels a longer distance. Observations also indicate that bimanual coordination is more carefully controlled when interacting objects are in close proximity or contact. In the pouring example, the kettle moves more than the cup during the phase pouring, but this may not hold true when considering both the reaching and pouring phases together.

Motivated by these observations, we propose an *invariance criterion*. First, we define the *object interaction period* as the time during which objects are either in contact or close by. During this period, we compute the spatial variation of both objects in all static candidate local frames (i. e., local frames anchored to static objects). The object exhibiting the lowest spatial variation in any such frame is designated as the master, while the other becomes the slave. This approach leverages statistical evidence to identify invariant task features, such as object roles.

More formally, given a pair of moving objects $(O_{\mathsf{A}}, O_{\mathsf{B}})$ with trajectories obtained via the perception pipeline in Section 4.1.4, we first detect the interaction period for each demonstration, denoted by $[t_0^n, t_1^n]$, where $n = [1, N] \cap \mathbb{Z}$ (see Appendix B.2). Then, we compute the *translational and rotational variability* of each object relative to all static local frames on $\mathcal{O}_{\text{static}}$. Without compromising generalizability, the j^{th} candidate local frame $\mathcal{F}_j$ is selected on a static object, and the following sections describe the detailed computations.

Translational variability

For a moving object O_l, we map the position trajectories of its candidate points to a common viewpoint, i. e., the local frame $\mathcal{F}_j$, and compute the translational deviation during the interaction period for each demonstration:

$$\mathcal{P}_{l,i,j} = \left\{ \mathbf{p}^n_{l,i,j}(t^n_1) - \mathbf{p}^n_{l,i,j}(t^n_0) \,\middle|\, n \in [1, N] \cap \mathbb{Z} \right\}, \tag{5.2}$$

$$\mathcal{P}_{l,j} = \left\{ \mathcal{P}_{l,i,j} \,|\, i \in [1, P_l] \cap \mathbb{Z} \right\}, \tag{5.3}$$

where P_l is the number of candidate points on O_l, and $\mathbf{p}^n_{l,i,j}(t)$ denotes the position of the i^{th} candidate point on O_l, expressed in $\mathcal{F}_j$ at time t during the n^{th} demonstration. We then apply PCA to the N deviation vectors in $\mathcal{P}_{l,i,j}$ to obtain the covariance matrix $\mathbf{\Sigma}^t_{l,i,j}$ for each candidate point. The translational variability is measured by the normalized trace of the covariance matrix (Dümbgen, 1998):

$$\boldsymbol{\eta}^t_{l,i,j} = \frac{\operatorname{tr}(\mathbf{\Sigma}^t_{l,i,j})}{d\varphi_l}, \tag{5.4}$$

where φ_l denotes the spatial scale of O_l. This normalization ensures that the score is independent of the object's size. The overall variability score for object O_l with respect to the local frame $\mathcal{F}_j$ is computed as

$$\boldsymbol{\eta}^t_{l,j} = \frac{1}{P_l} \sum_{i=1}^{P_l} \boldsymbol{\eta}^t_{l,i,j}. \tag{5.5}$$

Rotational variability

Analogously, we compute the rotational variability using the trajectories of candidate local frames estimated in Section 4.1.4. Mapping these trajectories to the local frame $\mathcal{F}_j$, we obtain the rotation deviation for each candidate local frame on O_l:

$$\mathcal{R}_{l,i,j} = \left\{ \mathbf{R}^n_{l,i,j}(t^n_1) \mathbf{R}^n_{l,i,j}(t^n_0)^\top \,\middle|\, n \in [1, N] \cap \mathbb{Z} \right\}, \tag{5.6}$$

$$\mathcal{R}_{l,j} = \left\{ \mathcal{R}_{l,i,j} \,|\, i \in [1, P_l] \cap \mathbb{Z} \right\}, \tag{5.7}$$

where $\mathbf{R}^n_{l,i,j}(t)$ is the rotation matrix of the i^{th} candidate local frame on O_l, expressed in $\mathcal{F}_j$ at time t during the n^{th} demonstration. We compute the covariance matrix $\mathbf{\Sigma}^r_{l,i,j}$ of the set $\mathcal{R}_{l,i,j}$ on the Riemannian manifold ($\mathbb{R}^3 \times \mathcal{S}^3$), and subsequently define the rotational variability score for each candidate local frame

as

$$\boldsymbol{\eta}^r_{l,i,j} = \mathrm{tr}(\boldsymbol{\Sigma}^r_{l,i,j}). \tag{5.8}$$

The overall rotational variability for object O_l with respect to $\mathcal{F}_j$ is then

$$\boldsymbol{\eta}^r_{l,j} = \frac{1}{P_l} \sum_{i=1}^{P_l} \boldsymbol{\eta}^r_{l,i,j}. \tag{5.9}$$

Master-slave relationship proposal

Since the units of translation and rotation differ, we first normalize the variability scores of all moving objects:

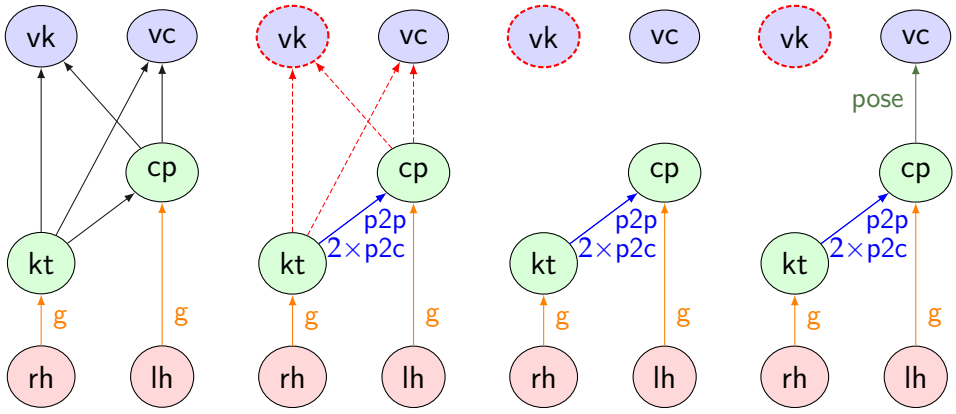
$$\hat{\boldsymbol{\eta}}^t_{l,j} = \frac{\boldsymbol{\eta}^t_{l,j}}{\sum_{l=1}^{|\mathcal{O}_{\text{moving}}|} \boldsymbol{\eta}^t_{l,j}}, \quad \hat{\boldsymbol{\eta}}^r_{l,j} = \frac{\boldsymbol{\eta}^r_{l,j}}{\sum_{l=1}^{|\mathcal{O}_{\text{moving}}|} \boldsymbol{\eta}^r_{l,j}}. \tag{5.10}$$

The master object for a moving object pair $(O_{\mathsf{A}}, O_{\mathsf{B}})$ is then identified as the one with the lower normalized translational or rotational variability observed in any static local frame. Formally, the master object and the corresponding local frame are jointly extracted by

$$l^*, j^* = \arg\min_{l,j} \left\{ \hat{\boldsymbol{\eta}}^t_{l,j}, \hat{\boldsymbol{\eta}}^r_{l,j} \right\}_{l \in \{\mathsf{A},\mathsf{B}\}}. \tag{5.11}$$

Since the variability scores are normalized and ranked across all moving objects, it is ensured that two objects cannot simultaneously act as the master with respect to one another. By exhaustively comparing all moving object pairs, we ensure that no bidirectional master-slave relationships occur. Moreover, each moving object is also considered as a potential slave to any virtual object. When a firm hand-object grasp is detected, the corresponding grasp relationship is established by assigning the object a master relative to the hand. Collectively, these steps yield a directed acyclic graph that captures the potential master-slave relationships.

For example, in the pouring water task shown in Figure 1.1 over eight demonstrations, the right hand is detected to grasp the kettle, making the kettle the master for the right hand, while the cup becomes the master for the left hand. Although both the kettle and the cup are moving, the cup is determined as the master relative to the kettle because its translational and rotational variability is lower. Subsequently, both the kettle and the cup are paired with their virtual



(a) Initial DAG (b) Truncation (c) Truncated DAG (d) HMSR

Figure 5.3.: The process of determining HMSR. (a) Starting from an initial proposal of DAG, (b) Uni-KVIL is applied to each link to derive the geometric constraints; (c) edges without constraints will be pruned; and (d) finally a pose constraint is added for the real cup relative to the virtual cup.

objects, yielding the initial DAG shown in Figure 5.3a. Redundant relationships in the initial DAG are pruned in the following truncation step.

5.1.5. Truncation

For each potential master-slave pair in the initial DAG, we employ Uni-KVIL to identify keypoint-based geometric constraints between them. For instance, when applying Uni-KVIL the kettle-cup pair in the DAG, we obtain a p2p constraint and two p2c constraints with the cup acting as the master, similar to the result presented in Uni-KVIL (see Figure 4.23d). No constraints are detected in other object pairs (indicated by ⇢ in Figure 5.3b). We then remove all pairs without any constraints, resulting in a more compact HMSR graph, where each remaining master-slave pair is supported by subsymbolic geometric constraints. After truncation, only the two hand-object pairs and the kettle-cup pair remain in the DAG (see Figure 5.3c).

As the number of demonstrations increases, the HMSR graph becomes increasingly compact and converges, a phenomenon discussed further in Section 5.4. With limited demonstrations, all salient geometric constraints must be considered due to the limited task knowledge; however, with more data, unnecessary constraints can be pruned based on accumulated statistical evidence.

When a moving master object O_{moving} is not constrained by any static object after truncation (e. g., the cup in Figure 5.3c), it is allowed to move freely in space following a task-space VMP. Training a VMP, however, requires demonstrated

trajectory data of a selected local frame on O_{moving}, and adaptation to a new scene necessitates modeling a *pose constraint* to derive the target pose for O_{moving}. In the absence of detected constraints for any candidate point on O_{moving}, we select a local frame based on the extracted task constraints. Since O_{moving} serves as a master object providing a local frame for its slave, we choose its candidate local frame $\mathcal{F}_{\text{moving}}$ that defines the highest priority constraints for the slave object. Given that O_{moving} shares the same initial state as its virtual object O^v_{moving}, the selected local frame is also defined on the virtual object (i. e., as $\mathcal{F}^v_{\text{moving}}$) and is used as the reference for mapping trajectories of $\mathcal{F}_{\text{moving}}$ into a common aligned viewpoint for VMP training.

Finally, we model the distribution of the selected frame $\mathcal{F}_{\text{moving}}$ at the final timestep T using a Gaussian Mixture Model (GMM) within the common viewpoint $\mathcal{F}^v_{\text{moving}}$. For task reproduction with a novel instance of O_{moving}, the selected frame is detected using the object canonical space described in Section 4.1.3, and a target pose is sampled from the learned GMM to adapt the VMP for execution. This constraint corresponds to the pose constraint illustrated in Figure 4.13f – an example of which, between the virtual cup and the real cup in the pouring water task, is shown in Figure 5.3d.

5.2. Bimanual Spatial Coordination Strategies

According to the bimanual manipulation taxonomy (Krebs and Asfour, 2022), bimanual tasks are classified into uncoordinated and coordinated categories. Unimanual manipulation –a subset of bimanual manipulation in which one arm is primarily used while the other remains idle –is included in both categories. The uncoordinated category is further subdivided into unimanual and bimanual uncoordinated tasks. In contrast, coordinated tasks are split into loosely and tightly coupled tasks based on the degree of interdependency between the arms. Loosely coupled tasks require only brief spatio-temporal coordination, whereas tightly coupled tasks are characterized by sustained physical interaction between the arms. Moreover, tightly coupled tasks can be symmetric, where both hands have the same role, or asymmetric, where one hand acts as a master, establishing reference frames for the other hand to execute the motion at different goals. This framework advances our ability to unify policy representations for both unimanual and bimanual manipulation.

Building upon this taxonomy, we propose a rule-based method to derive coordination strategies from the Hybrid Master-Slave Relationship (HMSR) described

in Section 5.1. In contrast to the approach in (Krebs and Asfour, 2022), our method learns bimanual coordination strategies from purely visual demonstrations; thus, the force information required to infer physical interaction is unavailable. Moreover, as the HMSR contains only spatial information, this chapter focuses on bimanual spatial coordination strategies, while temporal coordination is addressed in Chapter 6.

5.2.1. Uncoordinated Unimanual

Uncoordinated unimanual tasks involve scenarios in which a single arm is used. In the context of the HMSR, such tasks are represented by HMSR with a single hand leaf node. Common examples in daily activities include: 1) A single MSR with a hand-object pair (e. g., pressing a button or fetching tissue as shown in Figure 4.20 and Figure 4.21), where the hand directly manipulates the object. 2) A single-branch, multi-level MSR in which a hand grasps an object and manipulates it for tasks such as rearrangement relative to another object (e. g., hanging a hat in Figure 4.24, insertion in Figure 4.22, or pouring water in Figure 4.23) or uses it as a tool (e. g., cleaning a table as depicted in Figure 4.25).

More complex tasks may occur (e. g., mounting a part whose position is determined by two distinct mounting points, corresponding to a single-branch, three-level MSR with two master objects), or cases where an object's target is derived from multiple levels of master objects as in the Hybrid MSR. Although our framework supports a wide range of manipulation tasks, in this thesis we focus on common and simpler scenarios, leaving highly complex tasks for future work.

5.2.2. Uncoordinated Bimanual

In contrast to the unimanual case, the HMSR for uncoordinated bimanual tasks comprises two independent branches, each containing a single-hand leaf node. Each branch resembles the HMSR structure of uncoordinated unimanual tasks. For instance, each hand may grasp a different slave object and perform independent tasks, such as placing the objects at distinct locations. As illustrated in Figure 5.4a, the left and right hands place a spoon and a banana relative to different objects independently. In another example (Figure 5.4b), the left and right hands grasp the handles of a cutting board and a pan, respectively, without any constraints imposed between the arms during the grasping phase.

Figure 5.4.: Examples of different coordination strategies in daily tasks. In (a), an uncoordinated bimanual task is shown where the left hand places a spoon on a plate while the right hand independently places a banana on a tablemat. In (b), another uncoordinated bimanual scenario is depicted: the left hand reaches for the cutting board handle while the right hand approaches the pan handle, each operating independently. In (c), a loosely coupled coordination strategy is demonstrated; here, the left hand positions the plate beneath the spoon while the right hand lifts and then places the spoon back on top of the plate. (d) presents a case where the left hand positions the cutting board above the pan, and simultaneously the right hand places the pan onto a tablemat. Finally, in (e), tightly coupled symmetric coordination is illustrated by the simultaneous transport of a serving tray using both hands.

5.2.3. Loosely-coupled Coordination

Interaction forces between two *hand groups* (i. e., a hand and its grasped objects) are critical in distinguishing loosely coupled from tightly coupled asymmetric coordination strategies (Krebs and Asfour, 2022). Because estimating these forces from visual demonstrations is challenging, we do not differentiate between the two and instead group them under a single loosely coupled category. Specifically, if constraints exist between the objects grasped by each hand or if both grasped slave objects share at least one master object, the hand groups are classified as loosely coupled. In the first case, one hand is constrained by the other; in the second, both hands are directed toward the same master object.

For example, as shown in Figure 5.4c, a task in which a spoon is lifted from a table to place a plate beneath it requires modeling the virtual spoon at its initial location to derive the plate's goal. Here, the target placement of the spoon depends on the plate, and the two hand groups are spatially coordinated by the geometric constraints between the spoon and the plate. Similarly, in a transport task (Figure 5.4d), a cutting board must be positioned above a pan opening, establishing spatial coordination between the hand groups. Even without considering physical interaction forces, these cases demonstrate the derivation of loosely coupled coordination strategies.

5.2.4. Tightly-coupled Symmetric Coordination

In tightly coupled symmetric coordination, the following conditions hold over a significant time window: 1) Both hands grasp the same object. 2) The distance between the hands remains approximately constant, with a change rate below a specified threshold. The shared object between two hand groups implies physical interaction between them via the common object, resulting in tightly coupled symmetric coordination. As shown in Figure 5.4e, bimanual transport tasks often require this strategy, which is typically characterized by synchronized motion of both arms.

In the following section, we introduce a controller framework capable of reproducing all coordination strategies using keypoint-based task representations. In Section 5.4, we detail how the HMSR and the corresponding coordination categories are extracted from visual human demonstrations of various manipulation tasks.

5.3. Bimanual Keypoints-based Admittance Controller

Based on the subsymbolic task representations in the HMSR graph (including keypoints, their associated local frames, geometric constraints, and movement primitives), we derive a compliant, torque-controlled bimanual keypoint-based controller capable of executing diverse bimanual coordination strategies.

Each robot hand's Tool Center Point (TCP) is controlled using an admittance controller. The target pose for each TCP is derived from the task goal assigned to the corresponding hand group. Initially, hand groups are identified based on the grasp relationships specified in the HMSR. Because Bi-KVIL's subsymbolic task representation for each object pair in the HMSR is defined relative to the corresponding master object, the TCP command is computed by cascading the control commands of each master-slave pair back to the leaf node (i. e., the TCP). As described in Section 4.3, the control command for each master-slave pair combines attraction and density forces acting on the keypoints belonging to that pair. This principle is extended to each hand group in bimanual manipulation. However, in contrast to KAC in Uni-KVIL, the constraints on the keypoints may now depend on a moving master object manipulated by another arm or on multiple objects in the scene, with the associated VMP trajectories represented in a moving local frame anchored to the other object. These steps collectively form

a bimanual control framework, termed *Bi-KAC* – a natural extension of KAC that handles bimanual coordination by propagating control forces through master-slave relations to determine the control force for each TCP and, consequently, its target pose via Admittance law.

Furthermore, based on the bimanual coordination category extracted from the HMSR, we can implement different coordination strategies in the underlying real-time compliance controller to ensure robust performance under external perturbations. For example, in symmetric coordination, the target poses of both TCPs are subject to a constant distance constraint, and when a static relative pose is detected, a fixed SE(3) transformation constraint is applied between them. In loosely coupled cases, the target pose of the dominant arm is expressed in the TCP local frame of the other arm. In uncoordinated or unimanual cases, the two arms are controlled independently using the admittance controller.

By integrating the keypoint-based control framework with the coordination strategies derived from the HMSR into a real-time compliance controller, we achieve a unified control scheme that supports both unimanual and bimanual manipulations across diverse coordination categories, while simultaneously satisfying the task constraints for each object pair. In the following sections, we demonstrate how Bi-KVIL and Bi-KAC capture a wide range of bimanual coordination strategies from human demonstration videos and extract distinct fine-grained execution styles for the same task.

5.4. Evaluation

We evaluate our approach on eight manipulation tasks:

1. pouring water (Po_{w}),
2. pouring beer (Po_{b}),
3. placing a spoon ($\mathsf{Pl}_{\mathsf{sp}}$),
4. placing a serving tray ($\mathsf{Pl}_{\mathsf{st}}$),
5. placing a spoon and plate ($\mathsf{Pl}_{\mathsf{sp,pt}}$),
6. placing a cutting board and pan ($\mathsf{Pl}_{\mathsf{cb,pa}}$),
7. placing a spoon and banana ($\mathsf{Pl}_{\mathsf{sp,ba}}$), and
8. cleaning a table (C_{ta}).

For each task, demonstration videos are recorded with either an Azure Kinect or a Stereolab ZED camera. Our perception pipeline then extracts 3D point trajectories for objects and hands, from which the HMSR and the corresponding coordination strategy are derived. Finally, the tasks are reproduced with Bi-KAC in novel scenes featuring categorical objects. Our evaluation focuses on Bi-KVIL's ability to: 1) extract a consistent HMSR from demonstrations with varied task styles, 2) capture fine-grained subsymbolic task representations, and 3) reproduce the learned tasks with intra-category generalization.

5.4.1. Task Extraction

For each task, we provide demonstrations in different styles and in varying numbers ❾, resulting in a total of 14 evaluations.

Place spoon on plate with different execution styles

In the $\mathsf{Pl}_{\mathsf{sp}}$ task, the motion styles are defined as follows:

$\mathsf{Pl}^1_{\mathsf{sp}}$: The plate moves toward the initial position of the spoon, and the spoon is lifted and placed at the center of the plate.

$\mathsf{Pl}^2_{\mathsf{sp}}$: Similar to $\mathsf{Pl}^1_{\mathsf{sp}}$, but the plates originate from various positions above the table.

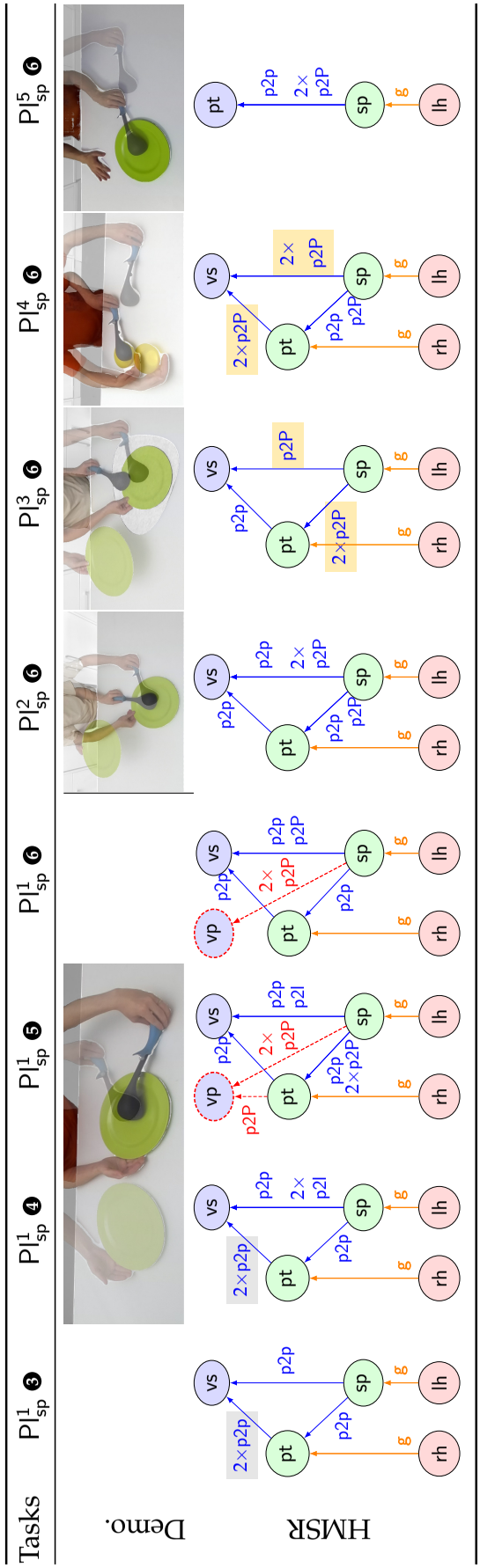
$\mathsf{Pl}^3_{\mathsf{sp}}$: As in $\mathsf{Pl}^1_{\mathsf{sp}}$, but the spoon is placed at an arbitrary position on the plate.

$\mathsf{Pl}^4_{\mathsf{sp}}$: The plate moves to an arbitrary position on the table while maintaining the goal of the spoon at its center.

$\mathsf{Pl}^5_{\mathsf{sp}}$: A unimanual placement scenario.

Results are presented in Table 5.1. With three and four demonstrations in $\mathsf{Pl}^1_{\mathsf{sp}}$❸ and ❹, Bi-KVIL extracts more p2p constraints (highlighted with) than in other tasks. This arises from the fact that the pose variations of plates relative to a virtual plate within four demonstrations are very small, which is not sufficient to estimate plane constraints. When additional demonstrations are provided in $\mathsf{Pl}^1_{\mathsf{sp}}$ ❺ and ❻, p2P constraints for the spoon are extracted with respect to multiple master objects (i. e., the plate, virtual spoon, and virtual plate). Since the plate always remains on the table, the existence of multiple p2P constraints is expected.

When the plate begins above the table in $\mathsf{Pl}^2_{\mathsf{sp}}$ ❻ with six demonstrations, Bi-KVIL learns to eliminate redundant master-slave pairs associated with the virtual



Note: Adapted from Gao et al. (2024). © 2024 IEEE.

Table 5.1.: Loosely-coupled task: place spoon (PI_{sp}) on a plate with different styles. Objects include a spoon (sp), a plate (pt), and two hands (lh, rh). We prefix the letter v to the corresponding virtual object, e. g., vs stands for the virtual spoon.

plate. This includes the pair between plate and virtual plate, and between spoon and virtual plate, as well as the corresponding p2P constraints (depicted as ⋯), resulting in a more compact HMSR graph. This illustrates that when a virtual object is not necessary in defining task constraints, it will be pruned.

When the execution style changes, the corresponding type of constraints will probability also change. For instance, instead of always placing the spoon head at the center of the plate like in Pl^1_{sp} and Pl^2_{sp}, the spoon head is placed at an arbitrary location on the plate in Pl^3_{sp}. Given six demonstrations, Bi-KVIL truncates the p2p constraints between the spoon and its two masters originally extracted in task Pl^1_{sp} and Pl^2_{sp}, and creates of an p2P constraint as a replacement (highlighted by). This aligns with the demonstration, as the spoon head is placed on the plate's surface, which approximately forms a plane constraint. The far end of the spoon handle is also subject to a p2P constraint as it is placed with different angles making its candidate points scatter on a plane aligning table surface.

Similarly, in Pl^4_{sp} ❻, the plate is constrained solely by the table surface, causing the p2p constraints between the plate and virtual spoon to be replaced by p2P constraints (highlighted by).

When only one arm is used to position the spoon head at the center of the plate in Pl^5_{sp} ❻, the plate remains globally static. In this scenario, the spatial constraints between the plate and the spoon in the form of p2p and p2P are identified, with the plate serving as the master object in a single-branch, multi-level MSR structure. This configuration indicates a unimanual, uncoordinated strategy.

Excluding the redundant master-slave pair observed in Pl^1_{sp} ❺/❻ and the unimanual case in Pl^5_{sp} ❻, the HMSR graph maintains a consistent structure across different task styles while differing in the subsymbolic constraints that capture motion differences. Moreover, as more diverse demonstrations are provided, redundant relations and constraints are eliminated. For all bimanual Pl_{sp} tasks, Bi-KVIL extracts a loosely coupled bimanual coordination strategy by identifying constraints between the two hand groups, with the right-hand group deemed non-dominant because the plate functions as the master for the spoon.

Pouring with different styles

In the Po task, the motion styles are defined as follows:

Po^1: The cup is fetched from a far position on the table and is moved closer to the kettle for pouring. The cup is held upright during the pouring action, while the kettle tilts with different angles across eight demonstrations.

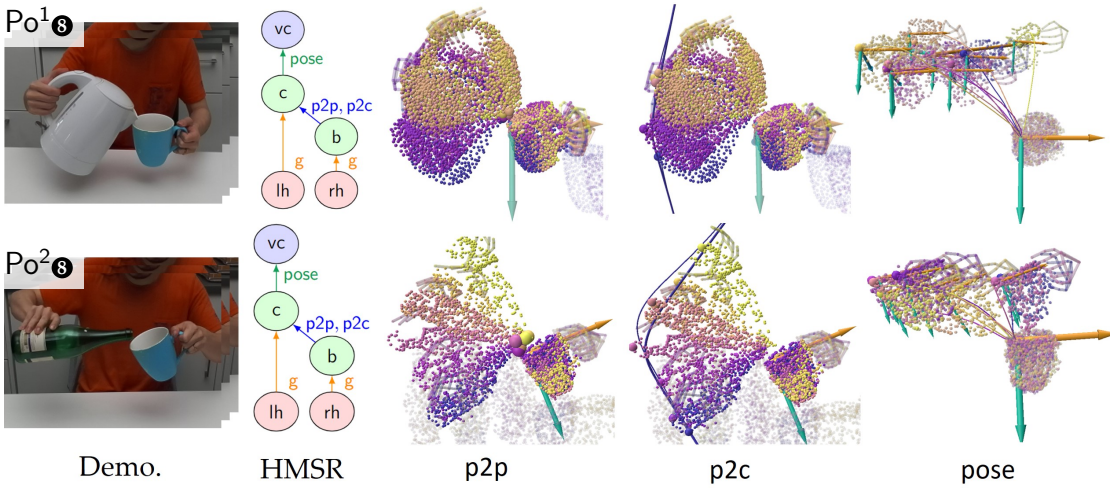


Figure 5.5.: Different styles of pour task. Given eight demonstration videos of each pouring style, we employ Bi-KVIL to extract the HMSR and geometric constraints. The corresponding p2p, p2c and pose constraints are illustrated in their corresponding local frames, respectively.

Po^2: In contrast to Po^1, the cup is placed initially close by the glass bottle, and tilts to a certain angle during pouring. Similarly, the glass bottle tilts with different angles across eight demonstrations.

As shown in Figure 5.5, the two pouring styles share the same HMSR structure at a symbolic level, differing only in the subsymbolic definitions of the p2p, p2c, and pose constraints. The p2p constraints in both cases are identified at the spout of the kettle and glass bottle, while the curve constraints p2c are located at the bottom of both objects. This result highly align with our intuition, showcasing the benefit of explicit representation of keypoints and geometric constraints. Notably, the pose constraints required to tilt the cup in the Po^2 task are captured by a GMM, while in Po^1 the GMM learns to generate upright cup poses. In both cases, the GMMs are represented in the virtual cup's local frame, which corresponds to the one defining the p2p constraints.

In both cases, the cup is designated a master object for the other object. It is important to notice that, in Po^1 task, the cup travels longer and faster on average than the kettle or the glass bottle during human demonstrations. In this case, the master-slave role assignment algorithm (Krebs and Asfour, 2022) relying on global motion saliency would fail, our pose invariance criteria presented in Section 5.1.4 correctly identify the cup as the master due to its reduced translational and rotational variability during the interaction period. This proves the effectiveness of the proposed method.

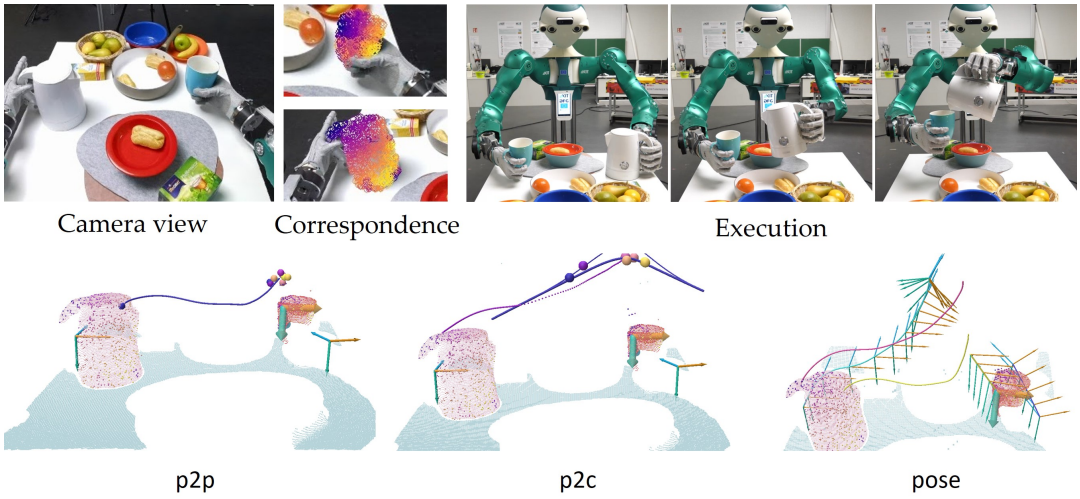
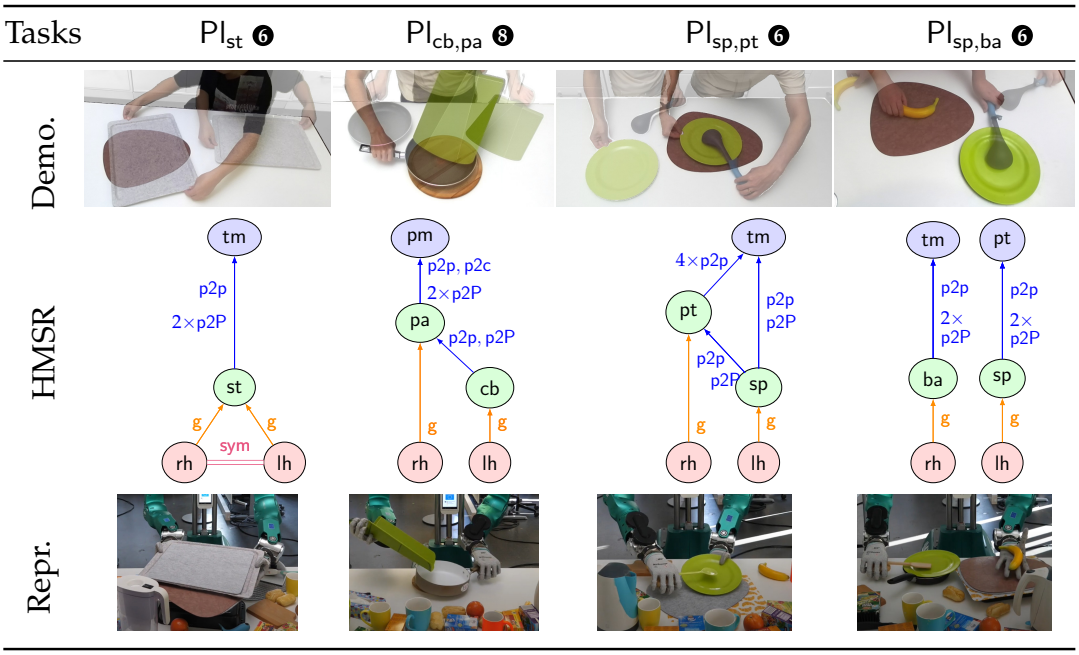


Figure 5.6.: The reproduction of the pouring water task using ARMAR-6. Given a RGB image from the robot perspective (top-left), we first perform the dense correspondence detection on both objects, i. e., kettle (kt) and cup (cp) to identify the keypoints on kettle and the corresponding local frames on the cup. Then we reconstruct the p2p, p2c and pose constraints. We execute the pouring action using Bi-KAC (top-right). After execution, we store trajectories of keypoints and local frames of both hands and the cup rim, and display them in the bottom-right image.

As shown in Figure 5.6, the left hand grasps the kettle following p2p and p2c constraints. The corresponding VMPs and constraints are dynamically updated based on the moving master cup grasped by the right hand, which is controlled by a task-space VMP aimed at a pose constraint defined on a static virtual cup. Importantly, as illustrated in the bottom-right image, the local frame attached to the cup's rim follows a VMP to move upward. Since the p2p and p2c constraints and their VMPs are represented in this local frame, they are continuously updated to reflect the upward motion. Therefore, the trajectory generated by the VMP for the first keypoint (p2p) in the initial frame (bottom-left) is updated over time, resulting in a global trajectory that combines the upward motion and its VMP trajectory.

Virtual objects

In both Po and Pl_{sp} tasks, the manipulated objects (e. g., kettle-cup, glass bottle-cup, and spoon-plate) are in motion during the manipulation, and virtual objects are maintained in the final HMSR representations as top-level master objects. For instance, the virtual cup in Po is initially truncated because no keypoint-based constraints are extracted; however, it is later appended to provide a pose



Note: Adapted from Gao et al. (2024). © 2024 IEEE.

Table 5.2.: HMSR for Pl_{st}, $Pl_{cb,pa}$, $Pl_{sp,pt}$, and $Pl_{sp,ba}$ tasks. Legend as Table 5.1 and tm, sp, pm, cb, pt, ba stand for tablemat, spoon, potmat, cutboard, plate, and banana. We show an example of a human demonstration (Demo.), the extracted HMSR, and an example of reproduction on ARMAR-6 (Repr.).

constraint (Section 5.1.5) for the moving master object, namely the real cup. In contrast, the virtual plate and virtual spoon remain in the representation since they provide essential keypoint-based constraints required by the task for either the real plate or real spoon.

It is important to note that virtual objects can be pruned when no constraints exist between them and the corresponding real objects. For example, the virtual plate is pruned in tasks Pl^2_{sp}, Pl^3_{sp}, and Pl^4_{sp}, while both the virtual kettle and the virtual glass bottle are pruned in the pouring tasks Po^1 and Po^2.

When a globally static real object is present, the virtual object may become redundant because the static object offers more pronounced spatial invariances for the moving objects under common viewpoints. In demonstrations with sufficient pose or shape variations, Bi-KVIL truncates the virtual objects in tasks Pl_{st}, $Pl_{cb,pa}$, $Pl_{sp,pt}$, and $Pl_{sp,ba}$, thereby allowing a static real object to serve as the top-level master (see Table 5.2).

For example, in task Pl_{st} ❻, Bi-KVIL extracts subsymbolic constraints p2p and p2P that define the target pose of the serving tray above the center of the table mat – a globally static object designated as the top-level master. A symmetric coordination is also extracted: in addition to the grasp constraint that ensures

both hands share a common serving tray, the near-zero rate of change in distance between the left and right hand suggest a prolonged period of physical interaction between the hand groups, constraining both hands to move synchronously with the serving tray.

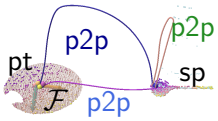
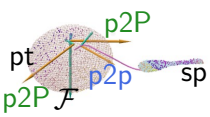
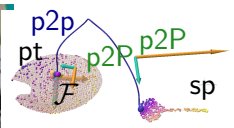
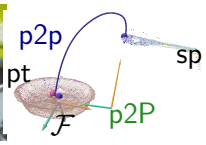
Bi-KVIL also accommodates tasks that involve more than two objects e. g., $\mathsf{PI}_{\mathsf{cb,pa}}$ ❽, $\mathsf{PI}_{\mathsf{sp,pt}}$ ❻, and $\mathsf{PI}_{\mathsf{sp,ba}}$ ❻). In task $\mathsf{PI}_{\mathsf{cb,pa}}$ ❽ (Figure 5.4d), the left hand places the cutting board above the pan such that its lower edge aligns with a point above the center of the pan, while simultaneously the pan is placed on the pot mat by the right hand. The master-slave relationship in the HMSR reveals that the left and right hand groups converge at the common object, the pan, indicating a loosely coupled coordination. Since the pot mat is globally static and provides constraints for the pan, virtual objects are not necessary for the task, thereby are pruned.

In contrast to task $\mathsf{PI}_{\mathsf{st}}$ ❻, no direct symmetric constraints are applied to both arms in the transport task $\mathsf{PI}_{\mathsf{cb,pa}}$ ❽, as they do not maintain a constant distance during the motion; here, the point-based constraints are utilized only for defining the temporal via points or the goal configuration of the objects. Similarly, task $\mathsf{PI}_{\mathsf{sp,pt}}$ ❻ requires both hand groups to achieve a common goal configuration defined by the top-level master object, the table mat, while the spoon in the left hand group is constrained by the plate in the right hand group. This configuration also results in a loosely coupled coordination.

In task $\mathsf{PI}_{\mathsf{sp,ba}}$ ❻, the connections between the two hand groups are truncated when no salient geometric constraint is identified, leading to two independent branches in the extracted HMSR. This outcome indicates uncoordinated bimanual actions; in each branch, real objects (the table mat and the plate) serve as the top-level master objects, eliminating the need for a virtual object. During reproduction, the two arms are controlled independently to follow the trajectories of the corresponding keypoints on each hand group.

Master object at multiple levels

In Section 5.1, we established that a master object defines constraints solely for its direct slave, contrasting with the inheritance mechanisms typical in object-oriented programming. As illustrated by task $\mathsf{PI}^2_{\mathsf{sp}}$ ❻ in Table 5.1, the spoon is constrained by multiple master objects (e. g., the plate and the virtual spoon) at distinct hierarchical levels, with the virtual spoon also serving as the master for the plate. The three p2p constraints are also displayed in Table 5.3 – (*1st row*).

Task	RGB	TR	Start	End
$\mathrm{Pl}^2_{\mathrm{sp}}$ ❻				
$\mathrm{Pl}^3_{\mathrm{sp}}$ ❻				
$\mathrm{Pl}^4_{\mathrm{sp}}$ ❻				
$\mathrm{Pl}^5_{\mathrm{sp}}$ ❻				

Note: Adapted from Gao et al. (2024). © 2024 IEEE.

Table 5.3.: Reproduction of the $\mathrm{Pl}_{\mathrm{sp}}$ tasks with different styles corresponding to $\mathrm{Pl}^{2-5}_{\mathrm{sp}}$ ❻. Images in each row correspond to RGB perception from the robot's viewpoint, the task representation (TR), and the start and end of execution.

The p2p constraint between the plate and the virtual spoon is not propagated to the spoon; instead, the virtual spoon directly imposes a separate p2p constraint on the spoon within a different local frame. Similar behavior is observed in the MSR of other tasks.

5.4.2. Task Reproduction

We qualitatively evaluate Bi-KAC for each task described in Section 5.4.1. In particular, one representative example per style of the $\mathrm{Pl}_{\mathrm{sp}}$ task is selected to illustrate the performance of Bi-KAC when reproducing the learned task using the ARMAR-6 humanoid robot (Asfour et al., 2019) with categorical objects in cluttered scenes.

For each row of Table 5.3, we show the learned task representation (TR) adapted to the perceived RGB image from the robot's viewpoint. The start and end states of the execution are shown from a third-person perspective.

For the task $\mathrm{Pl}^2_{\mathrm{sp}}$ ❻ shown in the first row, the plate is driven to the initial position of the spoon due to the p2p constraints between the plate and the virtual spoon. Concurrently, the spoon head is positioned above the center of the plate as a

result of the p2p constraints between the spoon head and the plate, while an additional p2p constraint governs the vertical motion of the spoon relative to the virtual spoon. The depicted curves represent the reproduced VMP trajectories of these three keypoints in their respective local frames at the initial timestep. Although the trajectory of the p2p constraint appears to direct the spoon head toward the plate due to the object-centric viewpoint, the combined execution of all trajectories results in an approximately vertical motion; the horizontal components cancel out as the plate moves toward the virtual spoon and the spoon moves toward the plate.

In contrast, for the task $\mathrm{Pl}^4_{\mathrm{sp}}$ ❻, a keypoint on the spoon head (indicated by p2p) follows a VMP trajectory toward the plate center from above, while a second keypoint on the spoon handle adheres to a planar constraint. Simultaneously, the plate is subject to a p2P constraint that is approximately parallel to that of the spoon handle; consequently, the plate moves to its most probable position on the plane manifold as determined by its learned density function.

In $\mathrm{Pl}^3_{\mathrm{sp}}$ ❻, the p2p constraints between the spoon head and the plate in task $\mathrm{Pl}^2_{\mathrm{sp}}$ ❻ are eliminated due to various target positions on the plate rather than always at the center. Though both the plate and spoon can be placed on the table, making the initial position of the spoon head lying on the target plane constraint, the learned VMP for such a p2P constraint still captures the up and down motion of the spoon head and handle as the motion's projection on the orthogonal direction of the plane constraint is used to train the VMP. As a result of the reproduction, the spoon head is not necessarily placed at the plate center anymore, rather it is guided by the density function on the plane constraint.

Furthermore, the learned tasks generalize to a variety of objects, including spoons of different shapes, plates of varying sizes and colors, and even cases where cooking pans replace plates in $\mathrm{Pl}_{\mathrm{sp}}$ (see Table 5.3). Additional evaluation results are available on our website[1], which accompanies the published paper (Gao et al., 2024).

5.5. Conclusion and Discussion

In this chapter, we extended Uni-KVIL's keypoint-based task representation to bimanual manipulation by leveraging an object-centric representation and hierarchical scene decomposition, resulting in the new approach named Bi-KVIL.

[1]Website for Bi-KVIL: https://sites.google.com/view/bi-kvil

Specifically, we proposed a novel Hybrid Master-Slave Relationship (HMSR) that effectively determines master-slave relationships among objects and organizes them in a compact directed acyclic graph. This framework extracts keypoint-based subsymbolic task representations for each object pair and subsequently derives the corresponding bimanual coordination strategies from the graph. The HMSR covers the bimanual manipulation taxonomy (Krebs and Asfour, 2022) and enables unified keypoint-based controllers for both unimanual and bimanual tasks. By explicitly modeling master-slave relationships and geometric constraints in an object-centric manner, our representation is embodiment-independent, viewpoint invariant, and generalizes well to categorical objects.

Furthermore, when a task is demonstrated with various fine-grained execution styles, Bi-KVIL effectively captures these differences through keypoint-based constraints while preserving a consistent HMSR topology. As discussed in Section 4.4.2, Bi-KVIL updates the constraint types as more demonstrations become available and prunes erroneous or redundant master-slave pairs to yield a more compact HMSR for task representation.

Bi-KVIL enables the learning of bimanual task representations with fewer than eight human demonstration videos captured via RGB-D or stereo cameras, without the need for additional devices. In comparison, other bimanual imitation learning approaches require a significantly larger number of demonstrations, e. g., 20 to 50 (Zhao et al., 2023; Fu et al., 2024b), 2500 to 4700 (Xie and Chowdhury, 2020), or 256 to 4000 (Kim et al., 2021, 2024). Some approaches also depend on teleoperation data (Zhao et al., 2023; Fu et al., 2024b) or human pose data recorded via motion capture systems (Liu et al., 2022). In contrast to methods that model keypoints implicitly as discussed in Section 2.1.3, our task constraints are explicit and closely aligned with human intuition.

As noted in Section 4.5, both Uni-KVIL and Bi-KVIL assume that the action of interest has been pre-segmented. Since natural human demonstrations often comprise sequences of actions, it is necessary to temporally decompose the demonstrations and group semantically aligned segments to learn both bimanual spatial coordination and task constraints. While motion segmentation yields elementary actions in the temporal domain, it also facilitates the analysis of bimanual temporal coordination, an essential aspect of bimanual manipulation, which will be investigated in Chapter 6.

CHAPTER 6

Keypoint-based Segmentation, Bimanual Coordination and Grasping

Manipulation tasks in daily activities typically consist of multiple sequential subtasks, each of which can be decomposed into distinct phases characterized by goal-directed motions (Kroemer et al., 2021). For example, a pouring task (Figure 1.1c) may serve as a subtask within a higher level "making tea" activity. This pouring task can itself be segmented into several goal-directed motions —- such as reaching, grasping, lifting, pouring, placing, releasing, and retrieving the hand. By decomposing the task into smaller components, the problem of learning task representations becomes more tractable as fewer factors need to be considered simultaneously. Moreover, such hierarchical decomposition facilitates the modular representation of common action segments (e. g., reaching and grasping) that recur across different high-level tasks.

In this chapter, we tackle the fundamental challenges outlined by our third research question on *learning action sequences*. Although many approaches have been developed in robotics and human motion analysis for temporal task decomposition, their application to bimanual visual imitation learning introduces unique challenges. These include issues related to object representation, dependency on local frame selection, hyperparameter tuning, and inconsistent segmentation granularity (see Section 2.4.1).

To overcome these challenges, we propose a novel *keypoint-based hierarchical motion segmentation algorithm* in Section 6.1. This algorithm decomposes visual

demonstrations into fine-grained motion segments and subsequently merges them into primitive actions. These primitives provide a consistent granularity that Bi-KVIL leverages to learn subsymbolic task representations. Furthermore, the perception pipeline proposed in Section 4.1.4 preserves detailed object shape information via a set of candidate points and establishes local frame alignment using an object's canonical space. By normalizing motion characteristics with an object spatial scaling factor, our method uses threshold values that are independent of specific objects or tasks, thereby enhancing generalizability.

In Section 6.2, we integrate temporal coordination into the Hybrid Master-Slave Relationship (HMSR) representation by leveraging the motion segmentation results. This integration yields a set of HMSRs that encapsulate the *spatio-temporal task representations* of each primitive action. A notable contribution in this context is our *task-oriented grasping framework* presented in Section 6.3, which is trained using human demonstrated grasping data captured at the starting point of the grasp segment.

Related content has been partially published in Cai et al. (2024). The contributions of this chapter are summarized in Section 1.2.2.

6.1. Keypoints-based hierarchical motion segmentation

Unlike the pre-segmented human demonstration videos used in Chapter 4 and Chapter 5 – which capture individual actions – the videos in this chapter consist of complete action sequences. These videos are preprocessed using the visual perception pipeline described in Section 4.1.4 to extract the position and velocity trajectories of candidate points, as well as the pose trajectories of candidate local frames. In our object-centric framework, each point trajectory is projected onto the candidate local frames associated with other objects. Figure 6.1 illustrates an example of a transport task with a sequence of RGB images and the corresponding candidate points at each step.

The proposed *keypoint-based hierarchical motion segmentation algorithm* partitions continuous motion into tractable segments and then merges these segments into higher-level action primitives with consistent granularity – a consistency that is critical for Bi-KVIL to effectively learn subsymbolic task representations. Notably, our approach derives all contextual information solely from object categories and motion characteristics.

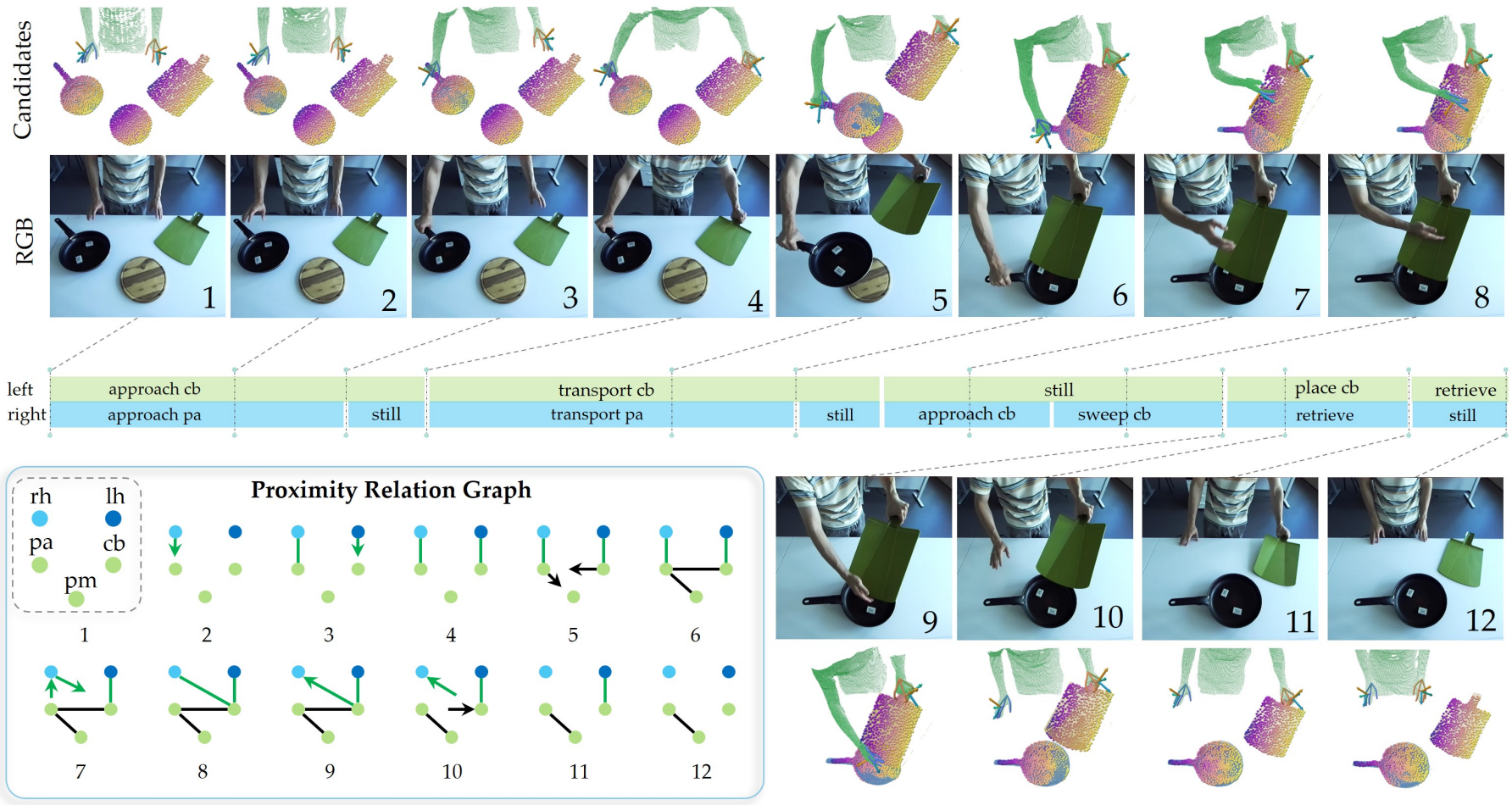


Figure 6.1.: A demonstration of a transport task. Twelve keyframes are shown, with RGB images, the candidate points on cutting board (cb), pan (pa) and potmat (pm), and human point cloud. The contact relation graph corresponding to each keyframe is displayed at the bottom left part of the image.

Because multiple objects (including hands) are typically involved in a task, the relative motion between each object pair provides valuable cues for identifying segmentation points and establishing the master-slave relationships discussed in Section 5.1.4. To this end, we first segment the motions for each object pair (see Section 6.1.1) and then merge the over-segmented elements into coherent higher-level action primitives through hand-group detection (see Section 6.1.2).

6.1.1. Proximity Detection

The hierarchical motion segmentation algorithms (Wächter and Asfour, 2015; Meixner et al., 2023) typically estimate only the object contact status. However, our experiments indicate that relying solely on contact events is insufficient in many scenarios. For instance, during a pouring task, the kettle and cup may not make direct contact if the pouring occurs from above the cup's rim. In such cases, contact represents a special instance of *static proximity* – where the distance between objects is nearly zero – while pouring from above reflects a different proximity status. Moreover, as demonstrated by Dreher et al. (2020b), dynamic spatial relationships (e. g., objects moving apart or converging) provide valuable cues for symbolic-level motion segmentation. We refer to the phase in which objects move apart as *divergent motion* (div) and the phase in which they converge as *convergent motion* (conv); both are classified under *dynamic proximity*.

Based on the relative distance and the nature of dynamic proximity before and after a static phase, we classify static interactions into four distinct categories: 1) *static contact*, 2) *static proximity*, 3) *static middle*, and 4) *static separation*, which are detailed in the subsequent sections.

Importantly, both static and dynamic proximity statuses are determined solely by motion characteristics, eliminating the need for manual labeling. To automatically identify the proximity status from the point-based representation of demonstration videos, we propose a three-step algorithm: *contact detection, global and relative motion saliency detection*, and subsequent *classification*.

Contact detection

For a given object pair $(O_\mathsf{A}, O_\mathsf{B})$, we analyze the position trajectories of candidate points to detect pairwise contact events at each time step, as described in Appendix B.2. Instead of using an absolute distance threshold, we adopt a ratio-based threshold relative to the spatial scale of the smaller or non-hand

object. Specifically, if O_A is the smaller or non-hand object with spatial scale φ_A, the distance threshold is defined as

$$\xi_{\text{dist}} = \hat{\xi}_{\text{dist}} \cdot \varphi_A,$$

and the corresponding threshold for the rate of change in distance is

$$\xi_{\text{dist,v}} = \hat{\xi}_{\text{dist,v}} \cdot \varphi_A,$$

where $\hat{\xi}_{\text{dist}}$ and $\hat{\xi}_{\text{dist,v}}$ are predefined ratio-based constants.

We define a *hover* status when the rate of distance change drops below $\xi_{\text{dist,v}}$ while the distance remains above ξ_{dist}, and a *contact* status when both values fall below their corresponding thresholds.

Global and relative motion saliency detection

The global motion saliency of an object is represented by the average norm of its candidate velocities in the global (i. e., camera) frame. We define a global static phase if the value drops below a threshold $\xi_{\text{g,v}}$, otherwise a global dynamic phase.

As noted in Section 5.1.4, the distance criterion alone does not capture the complete extent of relatively static information because it only considers the closest point pairs between objects. To overcome this limitation, we compute the velocity of the projected point trajectories of O_A across all candidate local frames on O_B. A larger spatial scale ensures a broader spatial extent, which in turn yields more accurate local frame tracking and improved relative velocity estimation. We quantify this by calculating a *relative motion score* as the average norm of the top-k largest velocities, thereby capturing the magnitude of prominent motions of O_A as observed from various perspectives on O_B. A static phase is then identified by applying the Zero-Velocity Crossing (ZVC, Fod, 2002) method to the relative motion score against a ratio-based threshold

$$\xi_{\text{r,v}} = \hat{\xi}_{\text{r,v}} \cdot \varphi_A,$$

where $\hat{\xi}_{\text{r,v}}$ is the relative velocity ratio threshold. If the score falls below $\xi_{\text{r,v}}$, the phase is considered relative static; otherwise, it is classified as relative dynamic. This frame-independent approach helps reduce over-segmentation. The mathematical formulation is provided in Appendix B.4.

To mitigate the effects of noise in the velocity signals, we employ a *soft-threshold* technique. If no significant relative velocity peak is detected during a phase initially classified as dynamic, that phase is reclassified as static.

Classification

Based on the relative static status of each phase, the proximity status is determined by its motion characteristics.

Relative dynamic proximity: Relative dynamic phases can be combined with different proximity statuses to model fine-grained dynamic proximity:

1. *Divergent motion* (div): A dynamic phase in which the average rate of change v of the top-k closest point pairs is positive.
2. *Convergent motion* (conv): A dynamic phase in which v is negative.
3. *Slide* (slide): A dynamic contact phase indicating that one object is moving along the surface of another.
4. *Dynamic hover* (hover_d): An overlap of dynamic and hover phases.

Relative static proximity: The relative static proximity status is determined by the surrounding dynamic proximity and the presence or absence of contact:

1. *Static contact* (cont_s): Occurs between a convergent and a divergent motion when the objects are in contact (i. e., the distance falls below ξ_{dist}).
2. *Static proximity* (prox_s): A static phase between a convergent and a divergent motion where the objects are not in contact.
3. *Static middle* (mid_s): A static phase occurring between either two convergent motions or two divergent motions.
4. *Static separation* (sep_s): A terminal static phase following a divergent motion or an initial static phase preceding a convergent motion.

Derived proximity status: Based on the above static and dynamic proximity statuses, we derive additional special proximity statuses commonly encountered in daily activities:

1. *Grasp* (grasp): A special type of static contact that additionally meets the enclosure criteria, i. e., when the fingers enclose any part of the object, as explained in Section 5.1.3.
2. *Touch* (touch): A static contact that does not meet the enclosure criteria.
3. *Static hover* (hover_s): An overlap of static and hover phases.

4. *Snatch* (snatch): A combination of a convergent motion phase directly followed by a grasp phase (conv–grasp), indicating that a hand moves toward an object, grasps it, and moves with the object without stopping.

These steps are applied to each object pair in every demonstration video, yielding the finest-grained motion segments based on proximity statuses.

Discussion

As discussed in Section 2.4.1, two major challenges emerge at this level: the *contextual labeling dilemma* and *granularity discrepancy*. Additional challenges arise in part from the varied demonstration styles that complicate the temporal alignment of semantically corresponding segments across multiple trials.

Granularity discrepancy: In Figure 6.2 steps #a–f, the cutting board is transported from the table to a position above the pan. During this motion, its distance to the pot mat increases and then decreases, producing segmentation points #a, #c, and #f in the last row of the plot. Notably, segmentation point #c marks the transition between a divergent and a convergent motion, indicating the furthest distance between the cutting board and the pot mat. Typically, the transport motion is learned as a continuous whole (from point #a to #f) to ensure a smooth trajectory. Therefore, segmentation point #c is not essential for learning constraints or motion representations, necessitating a merging step.

Furthermore, when segmenting motion based on trajectories of an object pair where one provides the local frames, simultaneous motion of both objects may result in a period of constant relative distance between either convergent or divergent phases. For example, as illustrated in Figure 6.2, a static middle phase (mid_s: steps #d–e) is detected between two convergent phases (conv: steps #b–d and #e–f), indicating that while the cutting board is moving closer to the pan, the distance remains nearly constant for a brief period. Such segments should ideally be merged to form a smooth and continuous motion.

Varied demonstration styles: Human demonstrations of the same task often exhibit diverse styles across different trials, resulting in varied proximity statuses for semantically identical actions. For example, the relative position of a kettle's spout to the cup rim may differ between demonstrations. If the distance in one demonstration exceeds the contact threshold due to a higher pouring position, the segmentation algorithm (Wächter and Asfour, 2015) may fail to register a contact event, thereby misaligning the pouring segments across trials. However, since both pouring styles are temporally located between a convergent and a

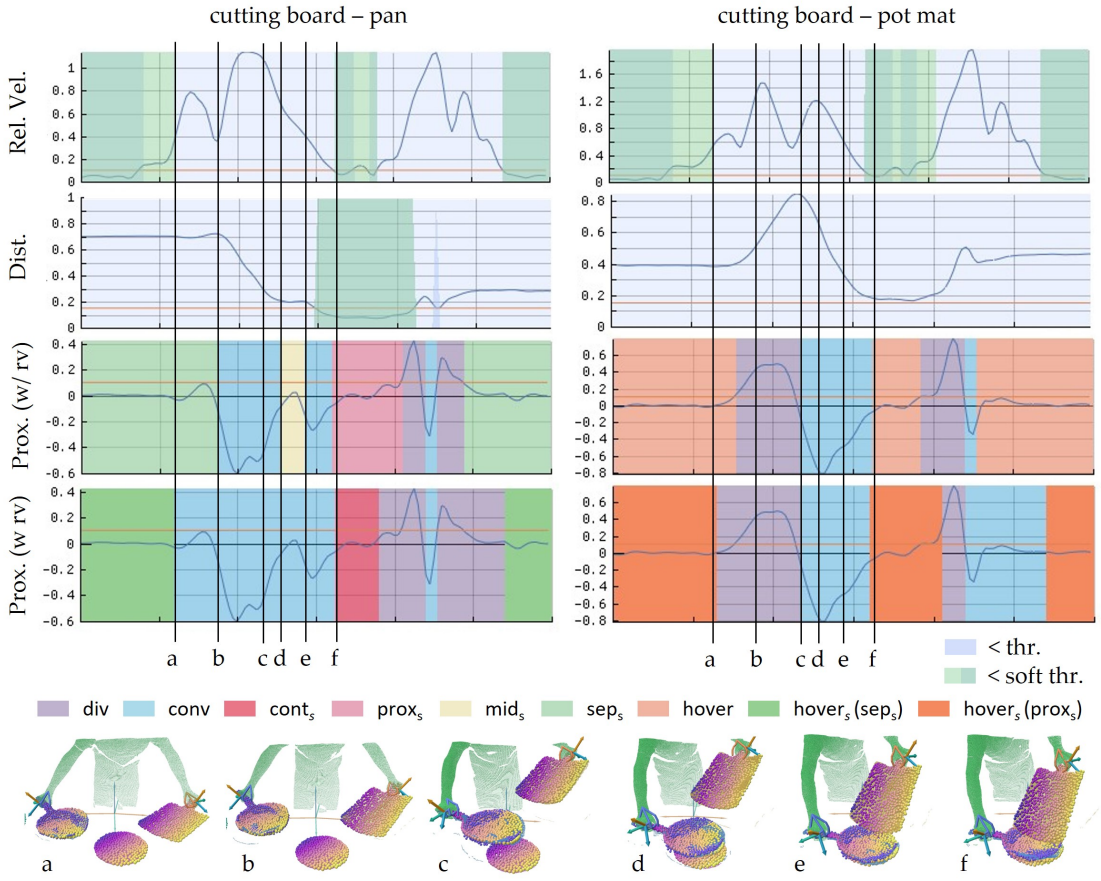


Figure 6.2.: Proximity status estimation. For each object pair, the relative velocity (Rel. Vel.), distance (Dist.), and the change rate of the distance are used to determine the proximity statues (Prox.). The change rate of the distance is plotted together with the proximity statuses (last two rows). The third row shows the proximity status estimated without relative velocity, while the last row with relative velocity. Pruning of the static middle (mid_s) is also shown in the last row.

divergent motion phase, the resulting static contact (cont_s) or static proximity (prox_s) segments can be aligned based on their contextual sequence. Leveraging the overall context of the detected proximity statuses enables effective grouping and alignment of segments from multiple demonstrations.

Temporal misalignment: The segmentation points are highly dependent on the execution of each demonstration and may not be consistently detected across different trials, leading to segments of varying granularity. This variability complicates the alignment of segments for learning Bi-KVIL's task representations. Moreover, although the sequence of actions is assumed to follow a consistent order across demonstrations, the precise timing for each arm's action can vary. For example, in Figure 6.1 (steps #1–4), the right hand grasps the handles of

the pan and before the left hand grasps the cutting board. In another trial, the left hand might grasp first. Such temporal misalignment, while indicative of temporal coordination of two arms, pose challenges for segment alignment and must be appropriately managed.

Segmentation point as via-point: Steps #6–11 in Figure 6.1 demonstrate a sequence in which the right hand reaches for the cutting board (#5–7), sweeps along its surface (#8–9), and then retracts (#9–11). As shown in Figure 6.3, the sweeping phase is characterized by a slide status between steps #b and #c, as a contact between the right hand and the cutting board is detected during relative dynamic motion. Although the continuous motion can be segmented and learned separately – with each phase's terminal hand pose serving as a target –this approach typically results in three discrete motions. When each motion segment is learned as a discrete motion individually, a motion blending technique is necessary to ensure smooth execution.

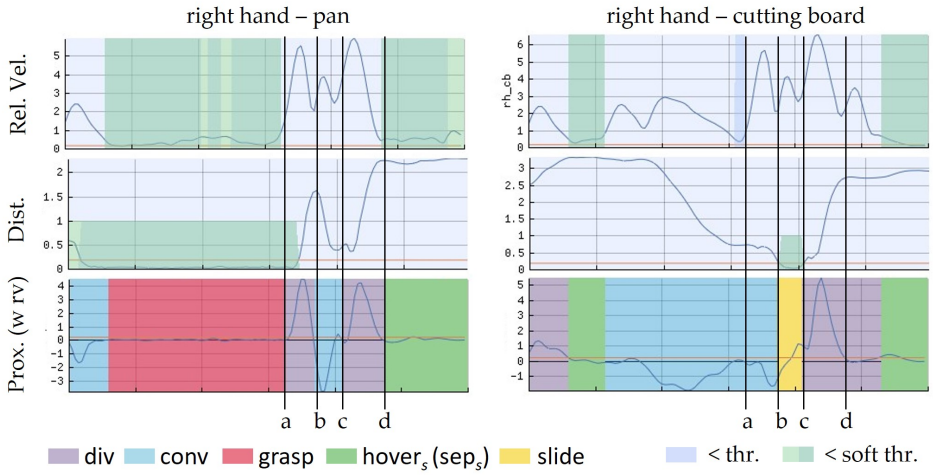


Figure 6.3.: Motion characteristics and proximity status for sweeping action.

Alternatively, considering that humans perform these actions continuously, it is reasonable to represent the entire motion as a unified sequence. In this framework, the hand poses corresponding to the points of contact initiation and termination serve as via-points. This strategy leverages the via-point adaptation capability of VMP (Section 3.3), which offers superior extrapolation compared to other movement primitive representations. Notably, since the slide status occurs between convergent and divergent motion phases (conv–slide–div), VMP can also be applied to learn dynamic hover motions if they follow a conv–hover$_d$–div sequence. Similarly, a snatch motion (snatch) can be modeled with the snatching target defined as a via-point relative to the object being grasped.

These types of actions are prevalent in daily activities such as sweeping debris on a table or vegetables on a cutting board, snatching a water bottle, spilling salt into a salad, or even playing tennis.

Contextual Labeling Dilemma: In Figure 6.1 (step #7), the right hand simultaneously leaves the pan handle while approaching the cutting board. The corresponding proximity detection results are illustrated in Figure 6.3 between steps #a and #b. When observed from the pan's perspective, the right hand appears to be diverging (div), whereas from the cutting board's perspective, it is converging (conv). Consequently, a single motion segment ambiguously encapsulates two distinct semantic meanings, depending on the viewpoint. This contextual ambiguity complicates the assignment of a unique semantic label to the segment when the context shifts. To address this challenge, the subsequent grouping technique must robustly incorporate these ambiguities, thereby supporting the learning of motion representations from multiple perspectives.

Next, we present a merging technique to address these challenges and to introduce the temporal coordination based on the segmentation results in Section 6.2.

6.1.2. Merging

Various merging algorithms have been explored in the literature (Section 2.4.1), such as merging temporal segmentation points using Gaussian Mixture Models (Prados et al., 2025) or combining similar segments based on their motion characteristics (Gutzeit and Kirchner, 2022; Tsai et al., 2019). However, these methods generally overlook the contextual information of over-segmented components, often resulting in inconsistent segmentation granularity. Furthermore, because motion segmentation for imitation learning tasks and the subsequent motion representation should be considered holistically, decoupling these problems may lead to misaligned motions that hinder robust motion representation. The degree to which motion segments should be merged depends on the representational capabilities of the motion model.

Hand group detection

We assume that a hand can only grasp one object at a time, though it may interact with multiple objects either by using the grasped object as a tool or via direct contact with other objects. This assumption imposes a priority among proximity

statuses for determining hand groups. Based on our observations of human activities, we define the following priority for the proximity statuses presented in Section 6.1.1:

$$\mathsf{grasp} > \mathsf{touch} > \mathsf{slide} > \mathsf{hover}_d, \quad \mathsf{cont}_s > \mathsf{prox}_s > \mathsf{mid}_s > \mathsf{sep}_s. \tag{6.1}$$

For each hand and for every hand-object pair associated with that hand, we first detect whether the hand exhibits a grasp (grasp), a static contact (cont_s), or a static proximity (prox_s) status with any object. If lower priority phases overlap a higher priority phase, they are pruned. We then group the remaining static phases (grasp, cont_s, prox_s) with the continuously connected relative dynamic phases that precede and follow them until the next static phase is reached. These grouped phases collectively constitute a valid hand group. Note that non-contact phases at the beginning or end are excluded from hand group detection because they lack sufficient contextual information. These steps are summarized in Algorithm 1.

For example, in the transport task shown in Figure 6.1, the left hand approaches the cutting board, grasps it, transports it above the pan, places it back, and finally retracts. As depicted in Figure 6.4, the left hand's grasp (grasp) of the cutting board is clearly detected, while lower-priority static hover (hover_s) phases between the left hand and the pan or pot mat during the grasp phase are pruned. Similarly, a static separation (sep_s) phase overlapping the beginning of the grasp phase is also pruned. With only the grasp phase remaining between the hand and the cutting board, the dynamic phases preceding and following it are grouped into a valid left-hand group represented as conv– grasp– div.

Similarly, for the right hand, static separation (sep_s) and static hover (hover_s) phases that overlap the grasp (grasp) phase with the pan are pruned. Grouping the dynamic phases around the remaining grasp phase yields a first valid right-hand group (conv– grasp– div) corresponding to reaching, grasping, transporting the pan, and releasing it. This group ends at step #c when the subsequent convergent motion conflicts with a slide (slide) phase between the right hand and the cutting board. By collecting the dynamic phases around the grasp phase and considering the imposed temporal limits, a second right-hand group is formed (conv– slide– div), which represents reaching for the cutting board, sweeping across its surface, and moving apart until the right hand remains still.

It is important to note that the proximity phases defined in Eq. (6.1) are not allowed to overlap; any lower-priority phases must be pruned when overlapping with a higher-priority phase. However, other proximity statuses may overlap.

Algorithm 1 Hand Group Detection Algorithm

Require: $\mathcal{S}_{\text{prox}}$: A set of proximity status sequences for a set of object pairs in a demonstration
Require: $\mathcal{H}$: Set of hands
Require: $\mathcal{O}$: Set of objects
Ensure: $\mathcal{G}_h = \{\mathsf{HG}\}$: Detected set of hand groups
$\mathcal{G}_h \leftarrow \emptyset$ ▷ Initialize hand groups
for all $O_h \in \mathcal{H}$ **do** ▷ Loop over each hand
  for all $(O_h, O) \in \mathcal{H} \times \mathcal{O}$ **do** ▷ Loop over all hand-object pairs
    $\mathcal{S}_{\text{prox}}(O_h, O) \in \mathcal{S}_{\text{prox}}$
    $\mathcal{S} \leftarrow$ detect_static_phases$(\mathcal{S}_{\text{prox}}(O_h, O))$ ▷ Detect grasp, cont_s, prox_s
    $\mathcal{S} \leftarrow$ prune_low_priority$(\mathcal{S})$
    $\mathcal{S} \leftarrow$ prune_non_contact$(\mathcal{S})$
    for all $s \in \mathcal{S}$ **do** ▷ Loop over all remaining static phases
      $\mathcal{G}_h \leftarrow \mathcal{G}_h \cup$ group_dynamic_phases$(s, \mathcal{S}, \mathcal{S}_{\text{prox}}(O_h, O))$
    end for
  end for
end for

function DETECT_STATIC_PHASES($\mathcal{S}_{\text{prox}}(O_h, O)$)
  return $\{s \in \mathcal{S}_{\text{prox}}(O_h, O) \mid s \in \{\mathsf{grasp}, \mathsf{cont}_s, \mathsf{prox}_s\}\}$
end function

function PRUNE_LOW_PRIORITY($\mathcal{S}$)
  return $\{s \in \mathcal{S} \mid s$ does not overlap with higher priority status$\}$
end function

function PRUNE_NON_CONTACT($\mathcal{S}$)
  return $\{s \in \mathcal{S} \mid s$ is not initial or terminal non-contact status$\}$
end function

function GROUP_DYNAMIC_PHASES($s, \mathcal{S}, \mathcal{S}_{\text{prox}}(O_h, O)$)
  $\mathcal{G} \leftarrow \emptyset$
  for all $d \in \mathcal{S}_{\text{prox}}(O_h, O)$ **do**
    $\mathcal{G} \leftarrow \{d \mid d$ is dynamic phase connected to s before reaching another $s' \in \mathcal{S}\}$
  end for
  return $\mathcal{G}$
end function

For instance, the right hand may exhibit a divergent motion (div) from the pan while simultaneously showing a convergent motion (conv) toward the cutting board, thereby addressing the contextual labeling dilemma presented in Section 6.1.1.

Hand Group Segmentation

Because our hand group detection relies on the relative proximity status between hand-object pairs, a static phase (e. g., a grasp phase) may consist of multiple sub-segments that need to be subdivided based on additional motion cues. For example, the left hand grasps the cutting board from step #4 to #11 in Figure 6.1

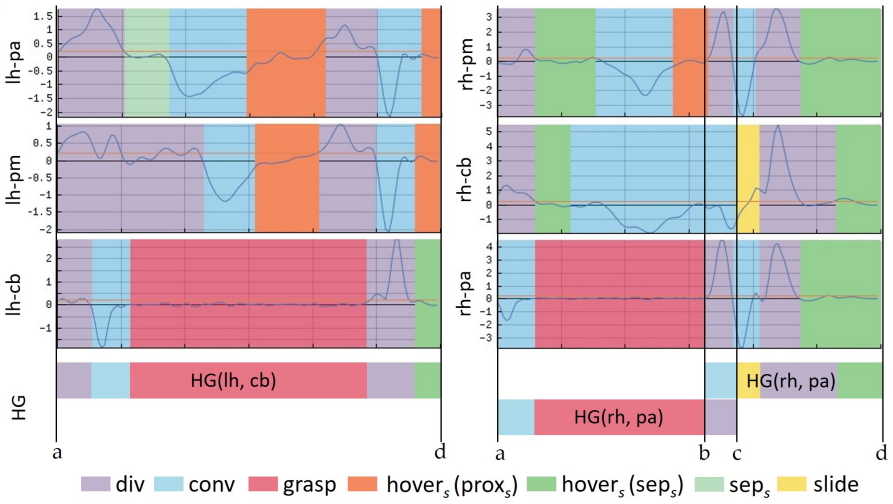


Figure 6.4.: Hand group detection for the transport task. Left hand forms a hand group with the cutting board HG(lh, cb) during the complete demonstration. Right hand forms two groups with pan HG(rh, pa) and with the cutting board HG(rh, cb). These two groups overlap at a dynamic phase.

(including moving the cutting board to the pan, holding it steady, and placing it back on the table). Learning the motion for the entire hand group may not be optimal; therefore, for each hand group, we further collect object-object segmentation points during the static phase if the object is associated with that hand group.

The objective of the *hand group segmentation* is to determine precise segmentation points for each hand group by leveraging relative dynamic phases and subdividing the static phase based on additional motion cues extracted from object-object interactions. To this end, we define three types of segmentation points based on their purpose and the confidence derived from motion cues:

1. A *hard segmentation point* is determined with high confidence and indicates the start or stop of a discrete motion.
2. A *weak segmentation point* is identified with lower confidence, but still indicates the start or stop of a discrete motion.
3. A *soft segmentation point* marks the initiation or termination of a slide, snatch, or hover$_d$ phase, and can be used to determine via-points.

In contrast to soft segmentation points, hard and weak segmentation points are typically used to define the definitive start and target points of a discrete motion.

We propose a five-step algorithm to detect these segmentation points.

Step 1 **Identification of Relative Dynamic Phases:** For a hand O_h, with $h \in \{\mathsf{left}, \mathsf{right}\}$, and a hand group $\mathsf{HG}(O_h, O_\mathsf{A})$ associated with object O_A, we first identify all relative dynamic phases in the proximity sequence of object pairs involving O_A (i. e., object pairs O_A–O_B) that overlap with the static phase of $\mathsf{HG}(O_h, O_\mathsf{A})$.

The insight here is that even if O_h is relatively static with respect to O_A, both may still be moving together when observed from a local frame of another object O_B with which O_A will interact. For example, from the local frame of the pan or pot mat, the left hand and the cutting board are observed to move together during steps #4–6 and #9–11 in Figure 6.1.

Step 2 **Merging of Successive Dynamic Phases:** Successive dynamic phases of the categories conv and div within $\mathsf{HG}(O_h, O_\mathsf{A})$ are merged to form a continuous, smooth trajectory for approaching and retracting motions. This step addresses the granularity discrepancy discussed in Section 6.1.1. Similarly, successive conv and div phases in the proximity sequences of each object pair O_A–O_B are merged.

Step 3 **Determination of Intra-Group Segmentation Points:** Within the detected hand group $\mathsf{HG}(O_h, O_\mathsf{A})$, each motion segment provides hard segmentation points at its start and end, with the exception of slide, hover_d, and snatch phases. These exceptional phases yield soft segmentation points that are used to specify via-points. If a hard segmentation point overlaps a soft segmentation point, the soft segmentation point is chosen.

Step 4 **Incorporation of External Segmentation Cues:** We then incorporate segmentation points provided by object pairs that lie outside the hand group $\mathsf{HG}(O_h, O_\mathsf{A})$. If a dynamic phase of the pair O_A–O_B (i. e., a phase during which O_A moves relative to O_B) overlaps with the static phase in $\mathsf{HG}(O_h, O_\mathsf{A})$, we first crop the dynamic phase to ensure it is completely enclosed by the static phase. We then evaluate how many objects O_B contribute segmentation information over the overlapping period:

1. In cases with a *single* such object, its start and end points are added as hard segmentation points if O_B is globally static within a time window $\xi_{\mathsf{sg,t}}$ around the segmentation point. Otherwise, they are added as weak segmentation points.

2. In cases with *multiple* such objects $\{O_\mathsf{B}\}$, we analyze the overlapping dynamic phases from different objects. The final decision is made by selecting the earliest hard starting segmentation point and the latest

hard terminating segmentation point. If both the starting or terminating segmentation points across these phases are soft segmentation points, their average is taken to define the start and end, respectively.

For example, when a hand grasps the kettle and remains relatively static during the pouring action, the start point of the hand group, as well as the critical point where pouring takes place, is determined by analyzing the kettle's motion relative to an external object, such as the cup.

Step 5 **Final Merging and Pruning:** Finally, overlapping segmentation points of the same type – often corresponding to the end of one segment and the start of the next – are merged by averaging their timestamps. Moreover, any weak segmentation point that lies within the time window $\xi_{sg,t}$ of a hard segmentation point is removed to ensure consistency and reduce redundancy.

The hand group segmentation algorithm is summarized in Algorithm 2. These five steps are repeated for each hand group, yielding refined motion segmentation points for each arm individually.

Summary

It is important to note that the keypoint-based motion segmentation algorithm produces segmentations at multiple levels of granularity. At the finest level, the proximity-based motion segments represent motion components that are coherent with respect to specific motion characteristics, such as maintaining contact and exhibiting convergent or divergent relative motion. Although these segments are derived purely from motion characteristics of object pairs, the resulting static and dynamic proximity statuses provide rich semantic cues for the hand group detection and segmentation algorithms. These cues enable the effective grouping and merging of fine-grained motion segments.

The proposed algorithms leverage both hand-object and object-object interactions to produce motion segmentation at the hand-group level and a more detailed sub-level, ensuring consistent granularity across segments. This merging technique not only addresses the challenges of contextual labeling and granularity discrepancy but also establishes a robust foundation for subsequent temporal coordination and constraint learning, as discussed in the following section.

Algorithm 2 Hand Group Segmentation Algorithm

Require: $\mathcal{H}$: Set of hands
Require: $\mathcal{O}$: Set of objects
Require: Hand group $\mathsf{HG}(O_h, O_\mathsf{A}) \in \mathcal{G}_h$
Require: Proximity sequences $\mathcal{S}_{\text{prox}}(O_\mathsf{A}, \cdot)$ of object pairs including O_A
Ensure: Precise segmentation points for each hand group

$\mathcal{O}_\mathsf{B} \leftarrow \{O \mid O \in \mathcal{O}, O \notin \mathcal{H}, O \neq O_\mathsf{A}\}$

Step 1: Identification of Relative Dynamic Phases
for each object pair O_A–$O_\mathsf{B}, O_\mathsf{B} \in \mathcal{O}_\mathsf{B}$ **do**
    Identify relative dynamic phases in $\mathcal{S}_{\text{prox}}(O_\mathsf{A}, O_\mathsf{B})$ overlapping with $\mathsf{HG}(O_h, O_\mathsf{A})$
end for

Step 2: Merging of Successive Dynamic Phases
for each identified dynamic phase in $\mathsf{HG}(O_h, O_\mathsf{A})$ and O_A–O_B **do**
    Merge successive conv and div phases
end for

Step 3: Determination of Intra-Group Segmentation Points
for each motion segment in $\mathsf{HG}(O_h, O_\mathsf{A})$ **do**
    Determine hard and soft segmentation points
end for

Step 4: Incorporation of External Segmentation Cues
for each dynamic phase of O_A–$O_\mathsf{B}, O_\mathsf{B} \in \mathcal{O}_\mathsf{B}$ overlapping a static phase in $\mathsf{HG}(O_h, O_\mathsf{A})$ **do**
    Crop dynamic phase to fit within static phase
    if single object O_B **then**
        Add start and end points as hard or weak segmentation points
    else
        Analyze overlapping dynamic phases from multiple objects
        Select earliest hard start and latest hard endpoints
    end if
end for

Step 5: Final Merging and Pruning
for each overlapping or close-by segmentation point **do**
    Merge by averaging timestamps if they have the same type
    if weak segmentation point within $\xi_{\text{sg,t}}$ of hard point **then**
        Remove weak segmentation point
    end if
end for
return Precise segmentation points for each hand group

6.2. Spatio-temporal Bimanual Coordination

In Chapter 5, we introduced the Hybrid Master-Slave Relationship (HMSR) to capture spatial constraints between object pairs and to describe the spatial coordination between two hands. However, this framework does not incorporate representations of temporal coordination. In this section, we investigate the temporal aspect of bimanual coordination, leveraging the motion segmentation results presented in Section 6.1. To achieve this integration, it is essential first to identify motion segments across multiple demonstrations that share identical semantic meanings (Section 6.2.1). By selecting one demonstration as a reference, we analyze the temporal coordination between the hand groups of both arms. The alignment of temporally corresponding segments, as detailed in Section 6.2.2, facilitates the extraction of spatial coordination and constraints via the Bi-KVIL framework. The result is a robust *spatio-temporal task representation* applicable to both unimanual and bimanual manipulation tasks. A qualitative evaluation of the proposed method using the transport task is provided in Section 6.2.3.

6.2.1. Semantic Alignment

Our approach enhances motion segmentation for bimanual actions by incorporating contextual information from the hand group segmentation algorithm presented in Section 6.1.2. Given the hand group segmentations for all demonstrations, we first perform an alignment at a coarse-granularity level, using the hand-object interaction relationships as strong indicators for proper alignment. Specifically, we detect all hand groups $\mathsf{HG}(O_h, O_i)$ in each demonstration using Algorithm 1 and establish a coarse alignment for hand groups corresponding to the same hand-object pair, O_h–O_i. For example, in all demonstrations of the transport task (Figure 6.1), the right hand initially groups with the pan to grasp and place it above the pot mat, and later groups with the cutting board during a sweeping action. This coarse alignment prevents the misalignment of motion segments that, despite exhibiting similar motion characteristics, belong to distinct hand groups.

Subsequently, motion segments within each hand group are aligned based on object-object interaction relationships. This process can be viewed as constructing object interaction graphs, where nodes represent objects and edges indicate different proximity statuses. For instance, as depicted in the bottom left corner of Figure 6.1, the five objects involved in the task are arranged in a graph that reflects their initial spatial arrangement in the RGB image at step #1. We use

arrows between object nodes to indicate a convergent motion if it points to another, and a divergent motion if it is reversed. A contact or static proximity status is established with a solid line connecting object nodes. In this way, we extend the contact relation graph of (Wächter and Asfour, 2015) and the spatial relation graphs described in (Ziaeetabar et al., 2018; Dreher et al., 2020b) using the proposed proximity statuses. This proximity relation graph robustly identifies semantically aligned motion segments within each hand group.

6.2.2. Temporal Coordination

By combining bottom-up segmentation and merging with top-down semantic alignment, we obtain aligned segmentation points for the hand groups of both arms. Using one arm as a temporal reference, we analyze the temporal distribution of the other arm's segmentation points across multiple demonstrations using a Gaussian Mixture Model. In this study, we adopt a simple temporal coordination criterion based on a predefined temporal threshold. For motion segments from two hand groups that overlap in time beyond the threshold ξ_{overlap}, we estimate the distribution of the temporal segmentation points – both at the beginning and end of the segments. We then compute the entropy and variance of these distributions and compare them to predefined threshold values. A consistently narrow time window for the temporal differences indicates that one event either consistently occurs simultaneously with, or a fixed time t seconds before or after, the corresponding event in the other arm.

Formally, we define *temporal coordination* as follows: Two motion segments from different hand groups are temporally coordinated if the temporal difference between their corresponding events (either initial or terminating points) consistently falls within a predefined narrow time window ξ_{overlap}. This definition encompasses cases where the events occur simultaneously or with a consistent fixed time offset.

In scenarios where motion segments in one arm correspond to globally static phases of the other, we consider them loosely coordinated. In temporally coordinated cases, we align the segments at their initial and terminating points and apply the Bi-KVIL framework to analyze the geometric constraints, thereby obtaining the spatial coordination as represented by the HMSR (Section 5.1). Both coordination types collectively represent the complete bimanual manipulation strategy.

6.2.3. Evaluation

We evaluate the effectiveness of our proposed keypoint-based hierarchical motion segmentation algorithm. We present a detailed analysis of the hand group detection and segmentation results, followed by an assessment of the semantic alignment process. Finally, we examine the temporal coordination between hand groups and summarize the overall performance of our approach in capturing robust spatio-temporal task representations.

Hand group detection and segmentation

Figure 6.5 illustrates the application of the *hand group detection algorithm* (Algorithm 1), which extracts the hand groups for each hand over different time periods. For instance, a hand group $\mathsf{HG}(\mathsf{lh}, \mathsf{cb})$ is identified in Figure 6.5a–(*top*), indicating that the left hand interacts continuously with the cutting board during the demonstration. At the onset of this hand group, a phase characterized by both divergent and convergent motion (div and conv) leads into the grasping phase (grasp), corresponding to a smooth, continuous approach action. However, this action is subdivided due to a change in the sign of the velocity of the relative distance. These over-segmented phases are subsequently merged using the *hand group segmentation algorithm* (Algorithm 2), resulting in a single dynamic phase corresponding to the approach action (steps #1–4 in Figure 6.1).

Rather than focusing solely on the object directly interacting with the hand, our approach also considers objects that interact with the manipulated objects, thereby providing additional motion cues and contextual information for more robust segmentation. For example, the interaction between the pan and the pot mat characterizes the overall motion of the left hand group during a relatively static grasp phase. Over-segmentation arising from these object-object interactions is also merged within the hand group segmentation algorithm, yielding a simpler composed motion segment and reducing redundancy in the segmentation points.

Similarly, for the right hand, two distinct hand groups are extracted: $\mathsf{HG}(\mathsf{rh}, \mathsf{pa})$ and $\mathsf{HG}(\mathsf{rh}, \mathsf{cb})$. These groups overlap during a dynamic phase, where the motion is recognized as divergent from the perspective of the pan and convergent from the perspective of the cutting board. This overlap resolves the contextual labeling issues discussed in Section 6.1.1. Furthermore, a sliding phase (slide) that represents the sweeping action is precisely captured. Soft segmentation points serve as via-points to ensure a smooth motion transition covering reaching,

sweeping, and retracting. Notably, weak segmentation points for the right hand group $\mathsf{HG}(\mathsf{rh}, \mathsf{pa})$, originally provided by the pan and cutting board pair, are eliminated when hard segmentation points from the pan and pot mat pair are available, thereby further reducing redundancy.

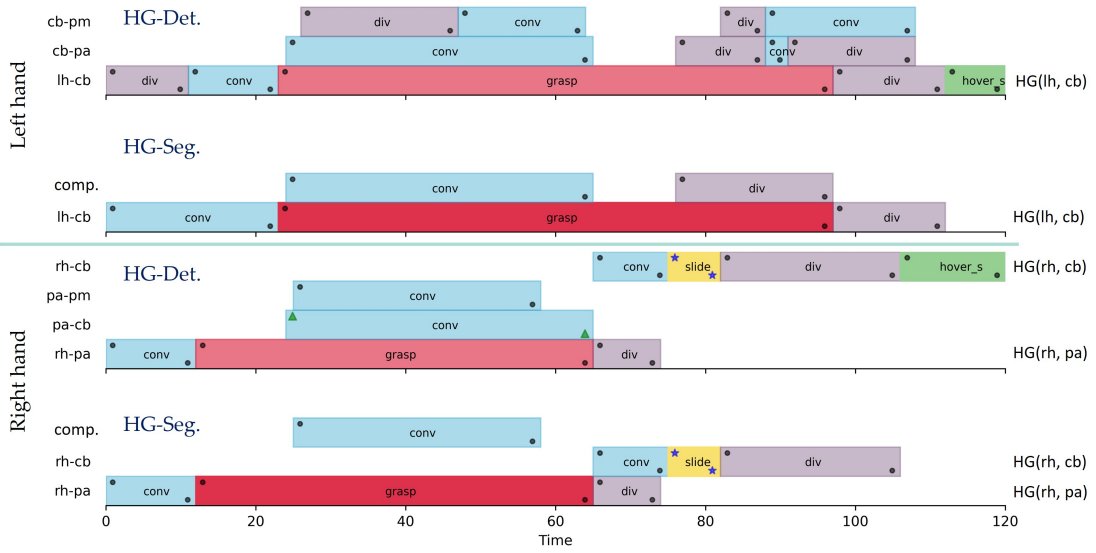
Figure 6.5b presents a second example of the transport task performed with a slightly different style and timing. As in the first demonstration, over-segmentation is merged and key proximity statuses are captured to extract consistent segmentation points. The motion segment structure and hand groups closely resemble those found in the first demonstration, allowing for consistent contextual information to establish robust semantic and temporal alignment of motion segments across different demonstrations.

Semantic Alignment

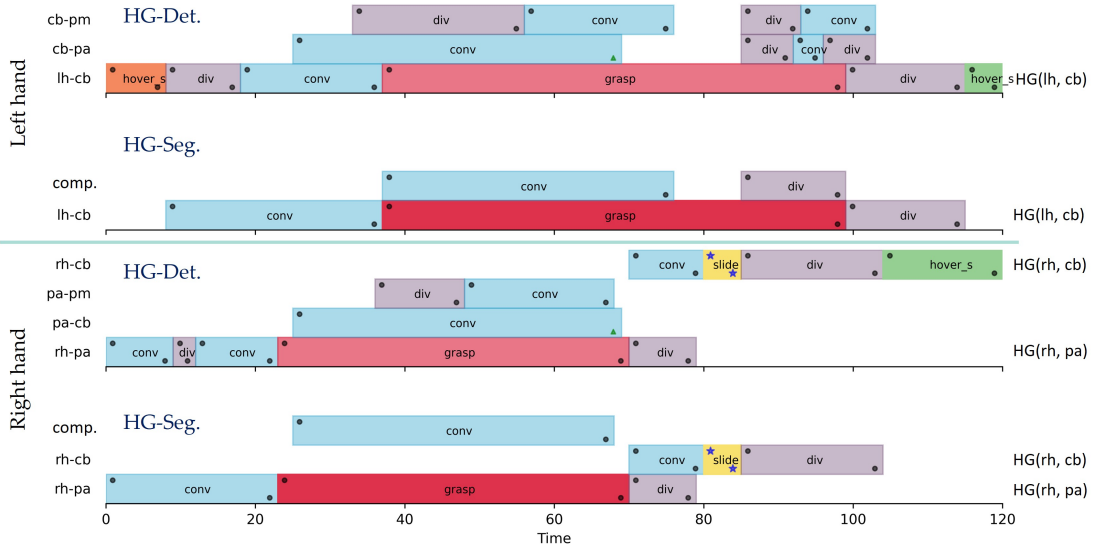
The proposed hand group segmentation algorithm yields a hierarchical temporal structure of the decomposed task. At the top level, hand groups provide a coarse alignment of motion segments, while at the finest level, proximity statuses facilitate detailed alignment. As illustrated in Figure 6.6, the light green area represents the aligned first right hand group $\mathsf{HG}(\mathsf{rh}, \mathsf{pa})$ across three demonstrations of the transport task, whereas the light blue region marks the aligned second right hand group $\mathsf{HG}(\mathsf{rh}, \mathsf{cb})$. This alignment is achieved by identifying the underlying hand-object interaction relationships.

Within each hand group, motion segments are further aligned based on their proximity types and object-object interaction relationships. As shown in Figure 6.7, our approach associates individual motion segments with their counterparts across different demonstrations. This alignment enables, for instance, the convergent motion at the beginning of $\mathsf{HG}(\mathsf{rh}, \mathsf{pa})$ – which corresponds to the action of approaching the pan handle – to be associated and learned as a movement primitive model. This model is then parameterized by a target grasp pose derived from the hand pose relative to the pan handle. The detailed motion representation and grasp learning will be discussed in Section 6.3.

It is important to note that the two hand groups corresponding to the same hand may overlap during dynamic phases, such as conv and div. In these cases, the same motion is captured from different perspectives, and the corresponding movement primitives are learned in different local frames (e. g., one relative to the pan and another relative to the cutting board).



(a) Demonstration 1



(b) Demonstration 2

Figure 6.5.: Hand group detection (HG-Det.) and segmentation (HG-Seg.) results for two demonstrations of the transport task are shown in Figure 6.1. In each segment, the corresponding proximity statuses are annotated at the center. The hard segmentation point (●), weak segmentation point (▲), and soft segmentation points (★) are marked in the top-left corner for the starting point and in the bottom-right corner for the terminating point. The hand-object and object-object pairs corresponding to each row are annotated on the left side of the plot, where lh, rh, cb, pan, pm, and comp represent left-hand, right-hand, cutting board, pan, pot mat, and a composition of multiple cues, respectively.

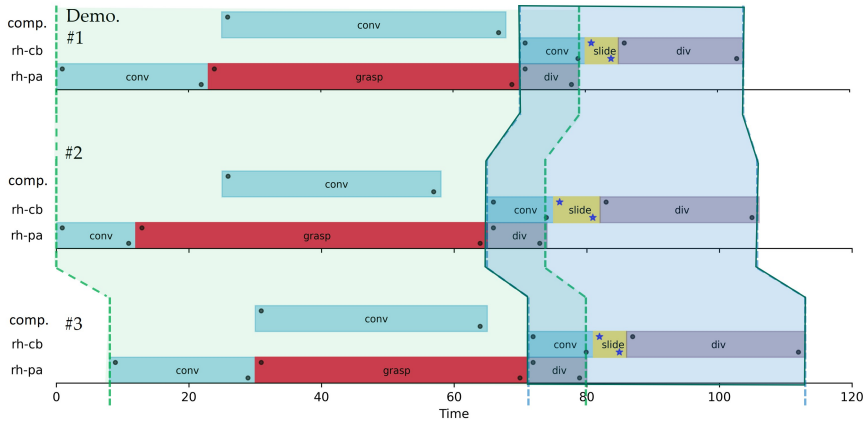


Figure 6.6.: Hand group alignment. Legend as Figure 6.5.

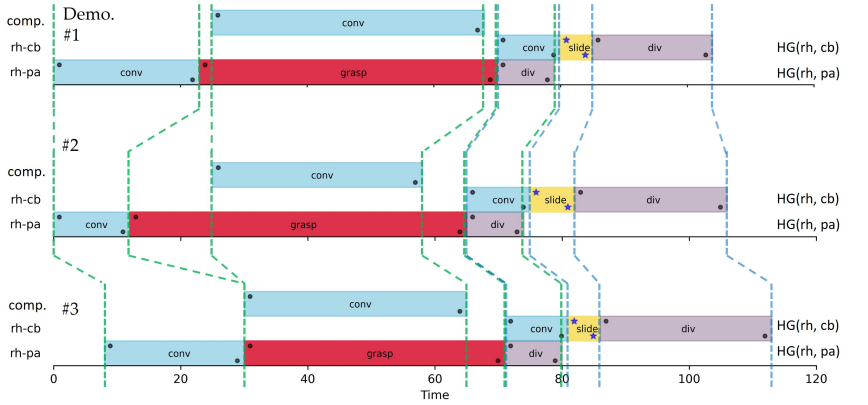


Figure 6.7.: Alignment of motion segments within a hand group. Legend as Figure 6.5.

Temporal Alignment and Coordination

As shown in Figure 6.5, the segmentation points for semantically identical segments are shifted in time across different demonstrations. For instance, the approaching action of the hand group $\mathsf{HG}(\mathsf{rh}, \mathsf{pa})$ is longer in the second demonstration than in the first. With seven demonstrations of the transport task, we extract the temporal differences of corresponding segments between the left and right hand groups and analyze their distribution.

Figure 6.8 displays examples of the extracted temporal difference distributions for the grasping and transport actions. The distribution for grasping (blue) indicates that, relative to the time when the left hand grasps the cutting board, the right hand most likely grasps the pan handle at time represented by the two modes. In contrast, the transport distribution shows that the pan placement event typically occurs at a fixed time offset relative to when the cutting board is placed above the pan. Notably, the grasping distribution exhibits two modes and a wider spread, suggesting that the grasping events of both hands may occur

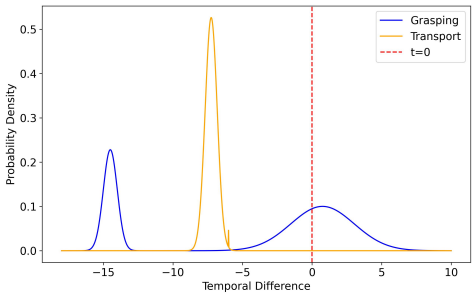


Figure 6.8.: Alignment of motion segments within a hand group.

independently. Conversely, the transport action demonstrates clear temporal coordination, with the pan transport consistently concluding approximately 7 steps earlier than the cutting board transport. This clear temporal coordination enables us to align the terminating points of the transport actions for both arms and to apply Bi-KVIL to extract the corresponding geometric constraints. We obtain identity HMSR as in task $\mathsf{Pl}_{\mathsf{cb,pa}}$ ❸ shown in Table 5.2, facilitating a *spatio-temporal bimanual coordination* representation.

Summary

We have presented a keypoint-based hierarchical motion segmentation approach capable of performing bottom-up segmentation solely based on motion characteristics and derived contextual information. This method provides consistent segmentation granularity across multiple levels, thereby facilitating the semantic alignment of motion segments across different demonstrations. Such alignment, in turn, enables the extraction of both temporal and spatial coordination using the Bi-KVIL framework. Evaluation of our approach on bimanual transport tasks shows promising results; similar performance has been observed in other tasks such as bimanual pouring with different styles (see Section 5.4). Future work will involve a comprehensive evaluation against state-of-the-art methods across various daily manipulation tasks.

Given that the segmented actions involve approaching an object and grasping it in a task-specific manner for subsequent manipulation, the next section will investigate task-oriented grasp learning using the data collected during the grasping segment. Other action segments, such as approaching, pouring, placing, and retracting, are also learned. These learned actions are then evaluated on a robot to execute a series of tasks for object grasping and rearrangement.

6.3. Task-oriented Grasping

In the previous section, we introduced a keypoint-based motion segmentation algorithm that exploits semantic context and motion characteristics to decompose human demonstration trajectories into semantically coherent segments. This segmentation enables the precise identification of critical events, such as the exact time points when grasping occurs. These instants capture essential hand configurations and pose information used for successful object manipulation, which we term *task-oriented grasps*.

In an object-centric framework, the distribution of demonstrated task-oriented grasp poses is modeled as a geometric constraint relative to an object (see Section 4.2) using a Gaussian Mixture Model (GMM; see Appendix B.6) defined on a Riemannian manifold ($\mathbb{R}^3 \times \mathcal{S}^3$). The learned grasp constraints are then transferred to categorical objects to ensure robust reproduction. To achieve this, we introduce a grasp learning and generation framework, as illustrated in Figure 6.9.

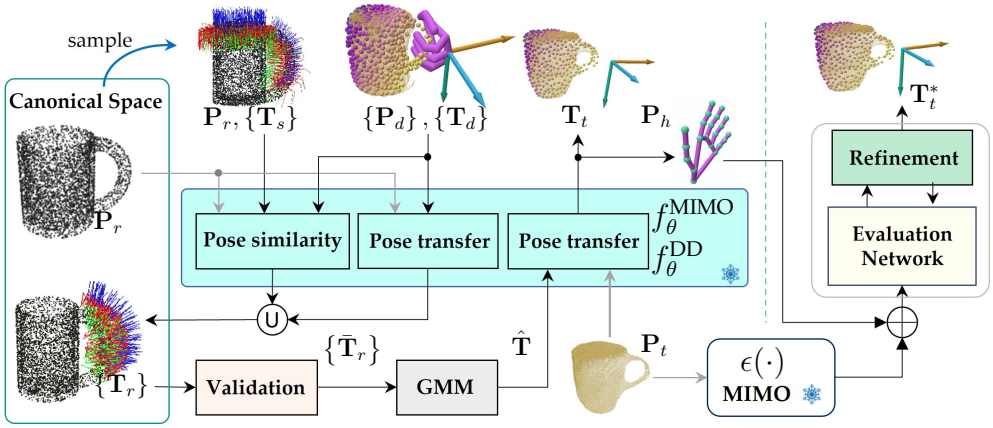


Figure 6.9.: The task-oriented grasping framework.

Grasp Learning

We assume that the object categories demonstrated by humans are known and that Multi-feature Implicit Model (MIMO) is pre-trained on these categories. Given an object canonical space as described in Section 4.1.3 and a set of demonstrated grasp poses $\{\mathbf{T}_d\}$ on corresponding object point clouds $\{\mathbf{P}_d\}$, we derive a set of task-relevant grasp candidates $\{\mathbf{T}_r\}$ in the canonical space using two complementary strategies:

1. We first generate a set of task-agnostic grasp candidates $\{\mathbf{T}_s\}$ using the method proposed in Sundermeyer et al. (2021) on the canonical point cloud $\mathbf{P}_r$. Next, we employ MIMO as a discriminator to select the grasp candidates from $\{\mathbf{T}_s\}$ that are most similar to the human demonstrations in $\{\mathbf{T}_d\}$ based on a pose similarity measure.

2. Alternatively, we directly transfer the demonstrated grasps $\{\mathbf{T}_d\}$ using 2D or 3D pose transfer (Eqs. (3.4) and (4.20)) to obtain candidate grasps relative to the canonical point cloud.

The candidate grasps $\{\mathbf{T}_r\}$ obtained from these strategies are subsequently validated in a simulation environment using Isaac Gym (Makoviychuk et al., 2021) with a humanoid hand to eliminate unstable candidates. The remaining candidates, denoted by $\{\bar{\mathbf{T}}_r\}$, are used to train a GMM, which is then employed to sample a target pose $\hat{\mathbf{T}}_t$ in the canonical space. This target pose is transferred to a novel object instance with an observed point cloud $\mathbf{P}_t$ via pose transfer during inference, resulting in a target grasp pose $\mathbf{T}_t$ for the testing object.

Grasp Evaluation and Refinement

To enhance generalizability, we propose a *task-agnostic grasp evaluation network* that predicts the robustness of executing a target grasp $\mathbf{T}_t$. This network is implemented as a Multi-Layer Perceptron (MLP) $\phi(\cdot,\cdot,\cdot)$ that takes as input the hand pose, the positions of its keypoints, and the latent code of the object point cloud obtained from a frozen encoder in MIMO. It outputs a grasp success probability:

$$\text{prob} = \phi(\mathbf{T}_t, \mathbf{P}_h, \epsilon(\mathbf{P}_t)) \in [0,1]. \tag{6.2}$$

The network is trained using a binary cross-entropy loss on a dataset that fuses task-agnostic grasp candidates across all objects and tasks, with binary labels indicating successful grasps as determined during the validation phase.

If the predicted grasp success probability falls below a predetermined threshold ξ_{grasp}, the grasp pose is refined by maximizing the grasp success likelihood according to the evaluation network:

$$\Delta\mathbf{T}_t^* = \arg\max_{\Delta\mathbf{T}_t} \phi(\Delta\mathbf{T}_t\mathbf{T}_t, \mathbf{P}_h, \varepsilon(\mathbf{P}_t)), \tag{6.3}$$

resulting in the optimal grasp pose $\mathbf{T}_t^* = \Delta\mathbf{T}_t^*\mathbf{T}_t$ for execution.

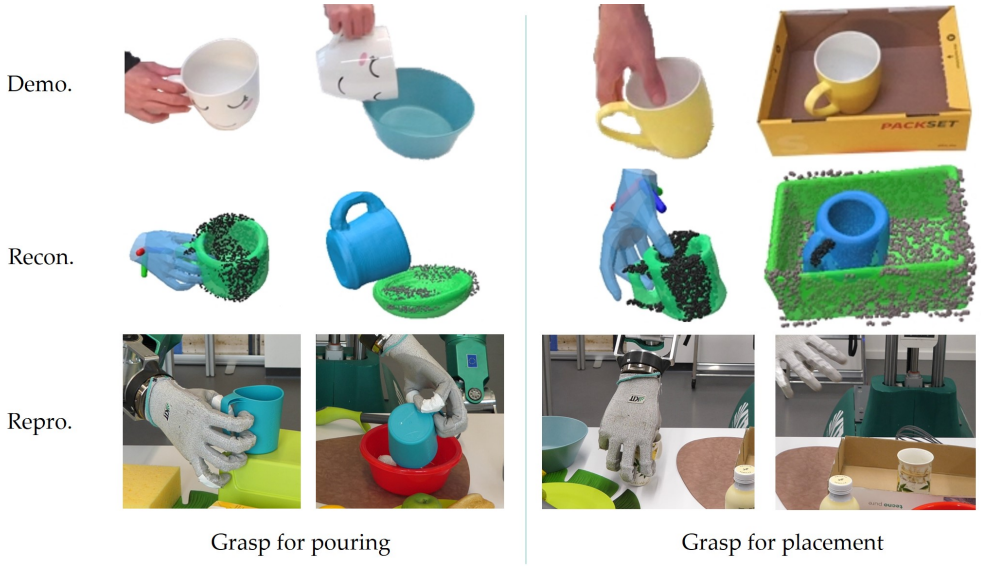


Figure 6.10.: Demonstration of task-oriented grasping, object reconstruction and reproduction.

6.3.1. Evaluation

The objective of the task-oriented grasping framework is to learn task-specific grasp constraints from human demonstrations and to generate corresponding target grasp poses for categorical objects. Notably, the same object may be manipulated differently depending on the task, thus the grasp constraint for each task may be anchored to a distinct functional part of the object category. In our evaluation, we define four tasks, where the mug and bottle are grasped differently for pouring and placement:

Po_m: Grasp a mug by its handle to pour into a bowl.

Pl_m: Grasp a mug by its rim to place it upright in a container.

Po_b: Grasp a bottle by its body to pour into a bowl.

Pl_b: Grasp a bottle by its neck to place it upright in a container.

Examples for the first two tasks are illustrated in Figure 6.10.

Grasping the mug handle for pouring

For the task Po_m, we provide a single demonstration that encompasses reaching, grasping, pouring, placing, and retracting phases. Human hand pose trajectories and object point clouds are obtained using the perception pipeline described in Section 4.1.4. Motion segmentation algorithms (Section 6.1) are then applied

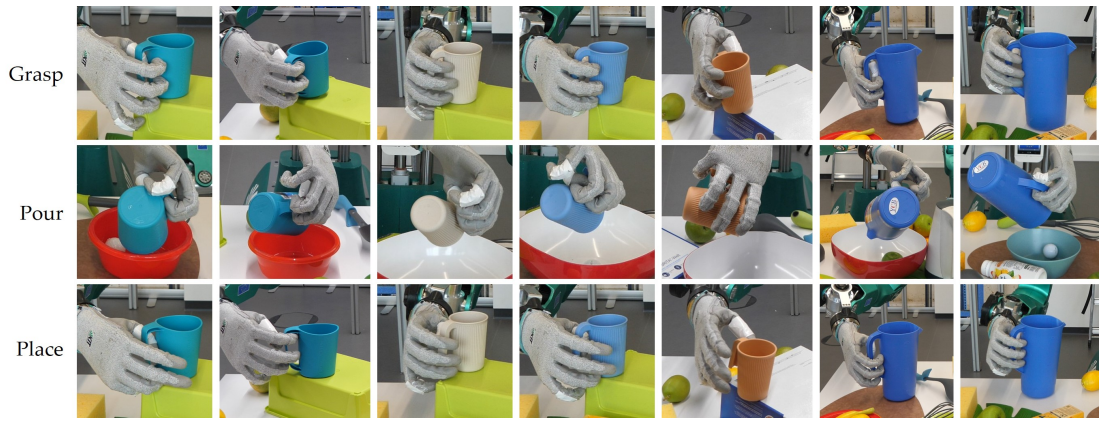


Figure 6.11.: Qualitative evaluation results of the grasp cup handle for pouring task. Keyframes at grasping, pouring and placement are shown for each trial.

to identify keyframes corresponding to grasping, pouring, and placing. These keyframes partition the hand trajectory into distinct phases (reaching, pouring, placing, and retracting), which are subsequently used to train Via-point Movement Primitives (VMPs, Section 3.3).

As detailed in Section 4.1.2, MIMO first reconstructs the object meshes from the partially observed point clouds (see Figure 6.10 – *2nd row*), which are then used to sample a point set for determining pose descriptors. For the grasping keyframe, the mug and hand are detected as a single MSR (Section 5.1) with the mug designated as the master object. The hand pose descriptor is computed by conditioning MIMO on the reconstructed point cloud of the mug. Similarly, for the pouring keyframe, the bowl is detected as the master object for the mug, and the pose descriptor of the mug is computed accordingly. For placement, we assume that the target hand pose remains the same as the initial grasp pose.

The object pose constraints between the mug and bowl are learned using a GMM, which allows direct sampling of the target pose for a mug relative to a new bowl during inference, similarly to the pose constraint of the cup in the pouring task described in Section 5.1.5. For the grasp pose constraint, a GMM is trained following the grasp learning procedure, and a refined target grasp pose is then obtained relative to a new mug instance. These target poses are used to adapt the corresponding VMPs for reaching, pouring, and placing motions.

A reproduction example for task Po_m using the humanoid robot ARMAR-6 is shown in Figure 6.10 – (*3rd row, left*). In this example, the robot successfully grasps the mug handle and executes the subsequent pouring action. The complete motion sequence, including the placement of the mug and hand retraction, is displayed in Figure 6.12 – (*1st row*). Multiple executions of each task are per-

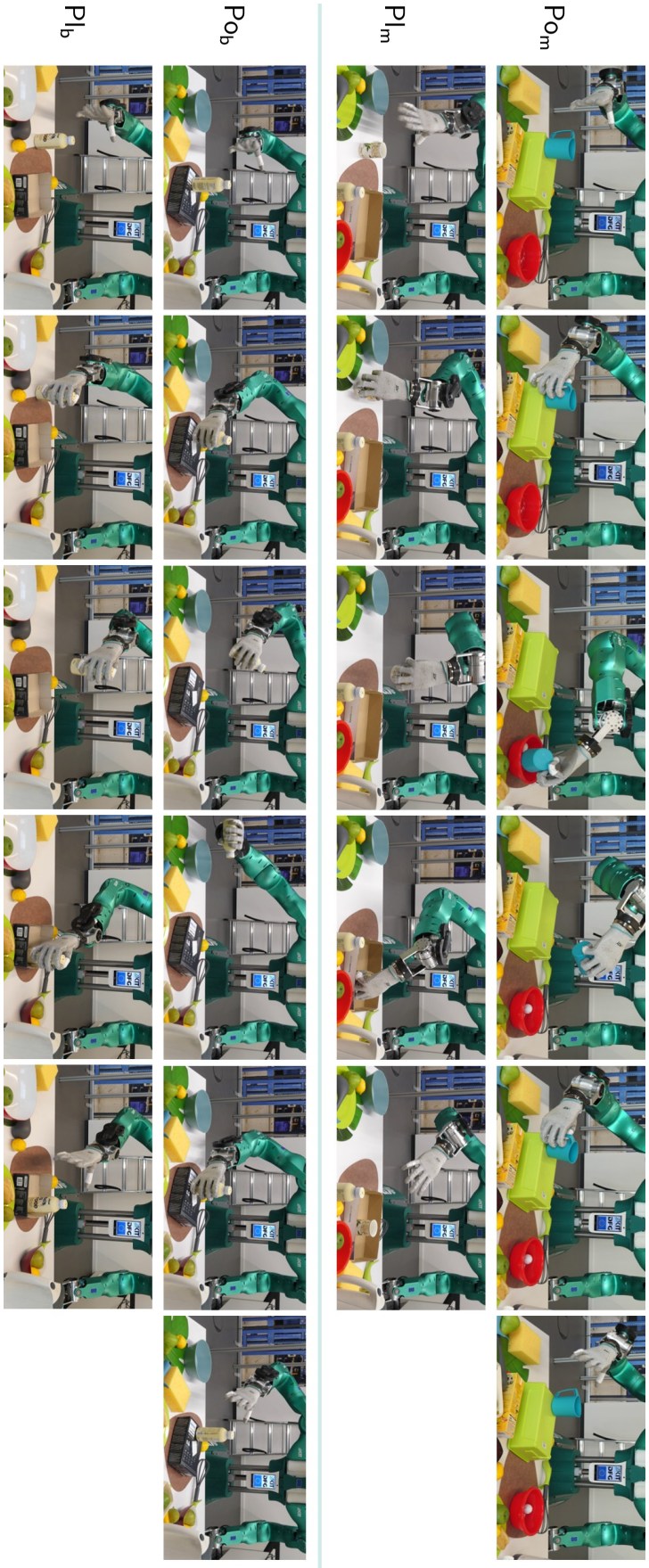


Figure 6.12.: Execution of the learned action sequence for pouring and placement with mug and bottle, individually.

formed with various categorical objects rearranged on the table (e. g., different cups and bowls as shown in Figure 6.11).

Grasping mug rim for placement

When placing a mug in a box, a side grasp on the handle may lead to a collision between the hand and the box. Consequently, a top grasp at the mug's rim is demonstrated (see Figure 6.10 – *right*). The evaluation procedure mirrors that of Po_{m}, except that following the placement of the mug inside the box, the robot directly retracts its hand as demonstrated. Successful execution examples are provided in Figures 6.10 and 6.12. Additional qualitative evaluation results for this task with categorical objects are detailed in Cai et al. (2024).

A similar evaluation is conducted for pouring and placement tasks using categorical bottles, bowls, and boxes. For the pouring task, the robot grasps the bottle's body, whereas for the placement task, it grasps the bottle's neck, following a single human demonstration per task. An example of robot reproduction is shown in Figure 6.12, with further results provided in Cai et al. (2024).

6.3.2. Summary

In this section, we presented a *task-oriented grasping framework* that learns task-specific grasp constraints modeled as Gaussian Mixture Models (GMMs). By leveraging the object canonical space provided by Multi-feature Implicit Model (MIMO), the framework effectively transfers and optimizes learned grasps and object poses to new instances of categorical objects. Notably, the superior ability of MIMO to distinguish fine-grained geometric details (Figure 4.6) enables the system to generalize from a single demonstration with only partial scene observations, a challenge where state-of-the-art methods such as Neural Descriptor Field (NDF, Simeonov et al., 2022a) and Neural Interaction Field and Template (NIFT, Huang et al., 2023) tend to fail. Quantitative evaluations presented in Cai et al. (2024) demonstrate that MIMO consistently outperforms NDF and NIFT across multi-view, single-view, multi-demonstration, and single-demonstration setups in simulated environments involving grasping and object rearrangement tasks.

Similarly to Dense Object Nets (DON, Florence et al., 2018), MIMO is trained per object category or per set of object categories, which presents challenges for deployment in open-vocabulary settings. Moreover, current 3D descriptor models (NDF, NIFT, and MIMO) have been evaluated exclusively on rigid objects for

point and pose correspondence detection and knowledge transfer. In contrast, the combined features of DINOv2 and DIFT (Section 4.1.1) offer unique capabilities for open-vocabulary tasks involving both rigid and deformable objects. The rich semantic information provided by DINOv2 features, which correspond to object functional parts, shows promise for generalizing subsymbolic task representations across object categories. However, 2D descriptors are inherently limited to detecting point correspondences on observable object surfaces. Although the pose transfer (Eq. (4.20)) leveraging 2D neural descriptors can also be applied to (grasp) pose transfer, its performance in inter-category scenarios has not yet been explored. In the context of visual imitation learning, we believe tackling object visual correspondence problem encompassing both 2D and 3D features in an open-vocabulary setting is a valuable research direction in the future.

6.4. Conclusion and Discussion

In this chapter, we proposed a *keypoint-based hierarchical motion segmentation algorithm* that consists of three main stages: *proximity detection, hand group detection, and segmentation*. This bottom-up approach relies on motion characteristics and derived contextual information to robustly merge and group initially oversegmented motion components into semantically meaningful longer segments, while preserving consistent granularity across multiple demonstrations. The resulting hierarchy—from high-level hand groups to fine-level proximity statuses – enables effective semantic and temporal alignment of motion segments, thereby facilitating the extraction of *spatio-temporal task representations*, including bimanual *spatio-temporal coordination*.

Building on this segmentation framework, we further developed a task-oriented grasping and generation framework. This framework effectively learns demonstrated grasps from a single or multiple demonstrations using partial scene observations and transfers the learned grasps to novel categorical objects. This is achieved by leveraging the Multi-feature Implicit Model (MIMO) architecture alongside a generalizable object representation based on object canonical spaces. The MIMO-based grasp learning and refinement framework demonstrates superior performance compared to state-of-the-art models in simulated and real environments, as evidenced across multi-view, single-view, multi-demonstration, and single-demonstration setups in grasping and object rearrangement tasks.

Although the proposed motion segmentation algorithms show promising results in reducing the manual segmentation effort previously required in the Uni-KVIL and Bi-KVIL frameworks – and are effectively applied in task-oriented evaluation scenarios – they have not been thoroughly evaluated in real-world long-horizon tasks. Furthermore, our current approach assumes that human demonstration videos consistently follow the same order in action sequences. In practice, this is not always the case, necessitating a more comprehensive temporal analysis, as discussed in Dreher and Asfour (2022).

Moreover, building a library of action primitives would enable the reuse of learned actions in new tasks, thereby enhancing the modularity and generalization capabilities of the imitation learning system. For instance, a learned pouring motion could be adapted from a "kettle-mug" task to a "milk box-bowl" task, as the objects used in the two tasks share similar functionalities and thereby should also resemble the task representations. Such transfer requires a similarity measure between task features, constraints, and motions. One promising approach is to measure object similarities at the part level by leveraging *affordance regions* (Hadjivelichkov et al., 2022) and transferring associated constraints across different object categories. Similarly, spatial relations (Kartmann et al., 2020) can capture task similarities, such as aligning the kettle's pout or milk box above the container, even when the objects and extracted subsymbolic constraints differ. These research directions are particularly important for achieving inter-category generalization, a critical capability for applying visual imitation learning systems in real-world scenarios, which will be considered in our future work.

CHAPTER 7

Conclusion and Future Work

The primary objectives of this thesis were to develop effective methods for extracting invariant task features – based on keypoints, geometric and temporal constraints, common viewpoints, and object-hand roles – from sparse human demonstration videos. In doing so, we established generalizable spatio-temporal task representations that unify unimanual and bimanual manipulation tasks. These objectives align with fundamental research questions in learning from human demonstration: *what to imitate*, *how to imitate*, and *when to imitate* (Billard et al., 2016). In this work, we focused on these three questions, assuming that a single demonstrator in the video is for imitation:

1) *What to imitate:* Identify and model invariant task features that are critical for task execution;

2) *How to imitate:* Develop statistical methods to extract these features as task representation; and

3) *When to imitate:* Determine appropriate temporal points for feature extraction to establish spatio-temporal coordination.

Inspired by the observational learning processes of human infants, we propose a bottom-up *Visual Imitation Learning* (VIL) framework that automatically extracts task features from human demonstration videos without relying on physical interaction or linguistic bootstrapping. The resulting task representation is interpretable, transferable, viewpoint-invariant, and embodiment-independent,

thereby significantly enhancing the generalization capability of VIL systems. We now summarize the core contributions of this thesis.

7.1. Contributions

In this thesis, we propose a novel *Keypoints-based Visual Imitation Learning* (KVIL) system that automatically extracts subsymbolic task representations from a small number of human visual demonstrations of both unimanual and bimanual manipulation tasks. The task representation consists of keypoints, their associated geometric constraints, local frames and movement primitives, as well as spatio-temporal coordination strategies. This system is designed to transfer across viewpoints, embodiment and intra-category object instances despite variations in object size, shape, and appearance.

7.1.1. Learning Keypoints-based Subsymbolic Task Constraints

The first major contribution is the development of Uni-KVIL (Chapter 4), a framework for learning keypoints-based subsymbolic task constraints for unimanual tasks. This includes: 1) A *neural-descriptor-based object representation* that enables dense correspondence detection, facilitating keypoint, pose transfer, and viewpoint alignment. 2) A *generalizable task representation* defined by keypoint-based geometric constraints on principal manifolds, their associated local frames, and movement primitives. 3) An efficient *Principal Constraint Estimation* (PCE) algorithm that jointly extracts all task representations from sparse human demonstrations. Notably, as more demonstrations are processed, the extracted constraints converge toward a more generalizable representation that better captures the task requirements. 4) A novel *keypoint-based admittance controller* (KAC) that prioritizes geometric constraints and enables successful task reproduction in novel scenarios. Evaluations on various manipulation tasks demonstrate that Uni-KVIL effectively extracts generalizable task representations in one-shot and few-shot VIL settings, while also exhibiting extrapolation capabilities by adapting to variations in object size, shape, and pose.

7.1.2. Learning Bimanual Coordination Strategies

Our second major contribution is the extension of the Uni-KVIL framework to learn bimanual coordination strategies, resulting in the Bi-KVIL system as presented in In Chapter 5. The principal contributions of this extension are: 1) A *Hybrid Master-Slave Relationship* (HMSR) model that structures the roles and relationships between object pairs. This model decomposes the scene by establishing master-slave relationships among objects and organizing them hierarchically in a compact directed acyclic graph. Each master object provides local frames and constraints for the associated slave objects. 2) The introduction of a *pose invariance criterion*, which leverages translational and spatial variabilities to identify motion-salient objects during interactions and designate them as slave objects. 3) The novel concept of *virtual object* to facilitate the modeling of task constraints relative to the initial states of the objects involved. 4) A rule-based system that automatically derives *bimanual coordination strategies* from the HMSR graph, thereby unifying unimanual and bimanual manipulation tasks under a single framework. 5) A *bimanual keypoints-based admittance controller* (Bi-KAC), which extends KAC with specialized control strategies for each coordination category. Control commands propagate through the HMSR graph to the robot end-effectors, enabling the reproduction of fine-grained bimanual tasks. The evaluation results demonstrate that Bi-KVIL successfully captures the fine-grained motion styles of the demonstrated tasks in the HMSR topology and the geometric constraints between each master-slave object pair. Moreover, the system can prune redundant master-slave pairs and constraints when they are no longer necessary.

7.1.3. Keypoint-based Segmentation, Bimanual Coordination and Grasping

The third major contribution of this thesis is the development of a framework that leverages the keypoint-based object representation to address motion segmentation, bimanual spatio-temporal coordination, and task-oriented grasping, as detailed in Chapter 6. Key contributions include: 1) A bottom-up *keypoint-based hierarchical motion segmentation algorithm* that decomposes bimanual demonstrations into fine-grained components based on the motion characteristics of object pairs and their proximity statuses. A subsequent hand-group detection and segmentation algorithm aggregates over-segmentations into action primitives of consistent granularity across multiple demonstrations, facil-

itating both semantic and temporal alignment. 2) *Spatio-temporal coordination strategies* for bimanual manipulation, where temporal coordination is modeled as a Gaussian Mixture Model based on motion segmentation results and spatial coordination is derived using Bi-KVIL. 3) A *task-oriented grasping learning and generation framework* based on Multi-feature Implicit Model (MIMO), which effectively learns demonstrated grasps in an object-canonical space and transfers the sampled target poses to novel object instances by leveraging the 3D neural descriptors provided by MIMO. This framework outperforms state-of-the-art approaches in single-view, multi-view, single-demonstration, and multi-demonstration settings.

7.1.4. Summary

In summary, this thesis presents a comprehensive approach to visual imitation learning by developing the KVIL system. Our work identifies several invariant task features that enhance generalization: 1) *sparse keypoints*, 2) *geometric constraints*, 3) *common viewpoints*, 4) *roles of objects and hands*, and 5) *temporal distribution*. To estimate these invariances, we employ robust and efficient statistical methods, including: 1) *Principal Component Analysis (PCA) and Principal Manifold Estimation (PME)*, 2) *Gaussian Mixture Models*, and 3) *a Pose Invariance Criterion*. Furthermore, we investigate the factors that contribute to the system's generalization and extrapolation capabilities, such as: 1) *neural descriptor-based object representation*, 2) *structured task representation*, 3) *keypoint-based geometric constraints on principal manifolds*, 4) *movement primitives with via-point adaptation*, and 5) *temporal decomposition of tasks*. The contributions made in learning keypoint-based subsymbolic task constraints and bimanual spatio-temporal coordination strategies establish a robust foundation for future research in visual imitation learning.

7.2. Outlook and Future Work

The thesis has laid a solid foundation for visual imitation learning by developing a comprehensive framework that addresses key aspects of task representation, bimanual coordination, and motion segmentation. Nevertheless, several promising research directions remain to further enhance the capabilities and practical applications of visual imitation learning systems.

7.2.1. Visual Correspondence

We have demonstrated that the 2D neural descriptors can effectively detect correspondences between categorical objects in open-vocabulary settings. However, without integrating techniques such as Thin-Plate Spline warping, the 2D descriptor model's feature space is insufficient for establishing smooth, dense correspondences. Although the proposed 3D descriptor model (MIMO) provides a smooth feature field and reliable correspondence detection around the object, it requires extensive training, making it challenging to apply in open-vocabulary settings. Future work will focus on developing neural descriptor models that integrate multiple data modalities to address these challenges. Compared to the current approaching of combining DINOv2 and DIFT features, unifying multiple data modalities in a single model can potentially also reduce the model size and inference time, facilitating online visual imitation learning. Importantly, because the neural descriptor models employed in this thesis are interchangeable, the subsequent keypoint and constraint extraction methods will not require modification.

7.2.2. Inter-category generalization via symbolic-level task representations

KVIL addresses the subsymbolic aspects of task modeling by extracting keypoint-based geometric constraints and generalizing them to categorical objects through visual correspondences. This generalization can be further enhanced by linking the extracted keypoints to symbolic task representations, such as affordances (Jiang et al., 2021). For example, keypoint neighborhoods in the semantic descriptor space may suggest affordance regions (Hadjivelichkov et al., 2022; Do et al., 2018), thereby enabling the transfer of a learned pouring task to similar tasks like pouring milk into a bowl. Moreover, spatial relations – commonly used to model the spatial distribution of affordance regions or object instances (Kartmann et al., 2021) – could facilitate transferring tasks such as "placing a bottle on the right of a bowl" to "placing a cup on the right of a plate." These semantic concepts improve the generalization of learned task representations across different object categories and tasks. We will consider integrating affordance and spatial relation representations to our VIL system as future work.

7.2.3. Articulated and Soft Object

The keypoint-based geometric constraints in this thesis are extracted based on a keypoint's position across multiple demonstrations at specific time points. In contrast, Sturm et al. (2011) presented a framework for learning linear and revolute articulated joints by leveraging the temporal trajectory of a point. Although our approach differs in the mathematical models used to represent geometric constraints, the similarities between the methods suggest that KVIL can be extended to estimate constraints for articulated objects. Future work will explore reconstructing articulation models from human demonstrations.

Furthermore, while the 2D neural descriptor models presented in Section 4.1.1 are applicable to both rigid and soft objects, their performance with soft and articulated objects has not been thoroughly investigated. Challenges may arise from severe occlusions and the numerous configurations of soft objects, making state tracking difficult without dynamic modeling. For instance, Huang et al. (2024b) demonstrated simple cloth folding tasks using DINOv2 features in a constraint-based optimization framework. It shows the potential of extracting keypoints on cloths placed flat on the table but does not cover more complicated soft object manipulation scenarios. Future extensions of the KVIL system will incorporate additional modeling techniques and constraints to effectively handle articulated and soft objects.

7.2.4. Social Learning

Psychological studies have shown that infants under two years old understand that evidence can be sampled in different ways, with each sampling process leading to different generalizations (Schulz, 2012). As noted by Gweon et al. (2010), samples may be drawn randomly from an entire population (weak sampling) or selectively from a subset (strong sampling). The selective sampling process, which reflects the demonstrator's preferences, remains a largely unexplored research area in robotics. Most task representation extraction methods focus on learning from samples while overlooking the significance of the sampling process. Moreover, learning the demonstrator's motion preferences is crucial for personalizing skills. For example, the varying pouring styles shown in Figure 5.5 may correspond to individual preferences, such as the desired cup tilt, while the evaluation in Table 5.1 suggests variations in spoon placement on a plate. Future research will investigate applying the KVIL framework to person-

alized skill learning and further explore the role of sampling processes in task representation.

7.2.5. Incremental Learning

While we have demonstrated that KVIL can update extracted geometric constraints with additional demonstration videos (e. g., in the pouring task shown in Figure 4.23 and the placing task in Table 5.1), the current approach employs a batch process using the Principal Constraint Estimation (PCE) algorithm. Future work will focus on developing incremental learning techniques that enable continuous updating and refinement of task representations as new demonstration data become available. Such incremental learning is critical for adapting to new tasks and environments without the need for complete re-extraction or extensive data storage, thereby enhancing the efficiency and scalability of visual imitation learning systems (Billard et al., 2016).

7.2.6. Reinforcement

Beyond updating task representations with new demonstrations, future research could explore the integration of Reinforcement Learning (RL) to further refine the learned geometric constraints and motion representations. By using the extracted task representations as a warm start, RL can significantly reduce the exploration space and improve learning efficiency. For instance, keypoint-based geometric constraints can serve as priors that guide an RL agent to focus on relevant areas of the state space, thereby avoiding redundant exploration. Additionally, the learned motion primitives can act as initial policies, which can be fine-tuned through RL to adapt to new environments or optimize specific performance metrics. This hybrid approach combining imitation learning with reinforcement learning holds great promise for enhancing the adaptability and performance of robotic systems.

Appendices

APPENDIX A

Computer Vision and Graphics

A.1. Recording of Human Demonstrations

For each task, we record a few demonstration videos (RGB-D) of a human performing the task using an Azure Kinect or a ZED 2i stereo camera mounted on a tripot in front of the demonstrator. The demonstrator is instructed to perform the same task with categorical objects varying in their shape, appearance, position and orientation on the table in each trial. The sequence of images are extracted from the recorded videos. When using Azure Kinect, we directly obtain depth images aligned with RGB images. In the case of ZED 2i stereo camera, we estimate the depth from the two stereo RGB images leveraging RaftStereo (Lipson et al., 2021), a deep learning model trained on Middlebury dataset (Scharstein et al., 2014). This approach accounts for transluscent, black and metal objects more robustly, where Azure Kinect tends perceive incorrect depth images.

For Chapter 4 and Chapter 5, we manually segment the motion period of interest corresponding to the task to be learned, while for Chapter 6, we provide the complete recording. In all cases, the demonstrations are resampled to have the same number of images, so that it is easier to leverage parallel computing for faster data processing. Since speed information is important for motion segmentation, we additionally provide the time stamps of each sampled image frame to the motion segmentation algorithm.

A.2. Hand Pose

The left and right hands's skeleton model with 21 keypoints corresponding to the MANO model (Romero et al., 2017) is shown in Figure A.1. The local frame of a hand is attached to the middle finger MCP keypoint, the orientation is defined with z-axis (→) pointing from palm to finger, x-axis (→) go through palm and y-axis (→) from thumb to pinky finger. The coordinate system for right hand is a right-handed coordinate system while for left hand a left-handed coordinate system. A mapping from left-handed to right-handed coordinate system is applied to have hand poses always represented as a right-handed coordinate system. Two grasp poses relative the kettle and cup handles are shown on the right-side of Figure A.1.

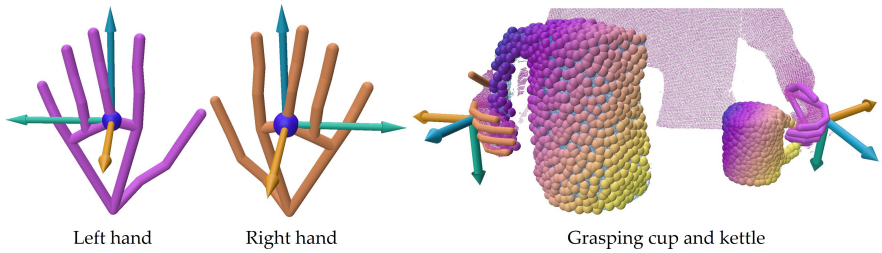


Figure A.1.: The definition of hand poses on left and right hand individually and an example of grasp poses around kettle and cup handles.

A.3. Viewpoint Augmentation

As illustrated in Figure A.2, we perform viewpoint augmentation on an image of an object by first detecting its binary mask. We then rotate both the RGB and mask images at various angles within the range $[0, 2\pi]$. Next, we crop the rotated images to obtain square patches centered around the object mask, ensuring that the object's bounding box occupies at least 80% of the cropped image area. Finally, we apply a horizontal flip operation to all the rotated images.

A.4. Multi-feature Implicit Model

A.4.1. Multi-task Loss Function

To train MIMO with four distinct feature branches, we combine the loss functions of each branch through a weighted sum. However, manually tuning these

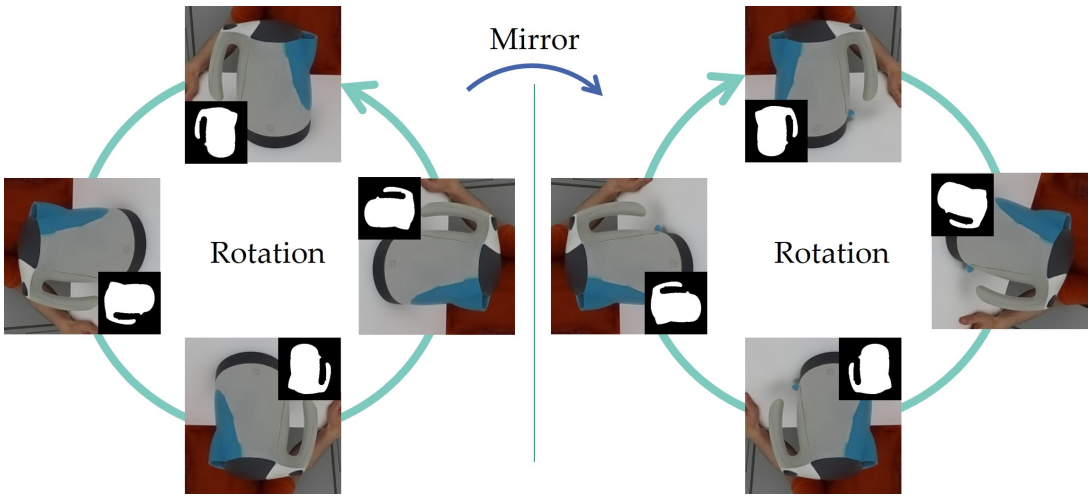


Figure A.2.: Viewpoint augmentation by rotation and horizontal flip.

weights is challenging. To address this problem, we introduce homoscedastic uncertainty (Kendall et al., 2018) for each branch, where the likelihood is defined as a Gaussian $p(\mathbf{y_i}|f_{\mathbf{W_i}}(\mathbf{x})) = \mathcal{N}(f_{\mathbf{W_i}}(\mathbf{x}), \sigma_i^2), i \in [1, 4]$ with the model output $f_{\mathbf{W_i}}(\mathbf{x})$ as the mean and the variance σ_i representing the uncertainty. The objective is to minimize the negative log-likelihood, i. e. $\mathcal{L} = \sum_{i=1}^{4}(\frac{1}{2\sigma_i^2}\mathcal{L}_i(\mathbf{W_i}) + \log(\sigma_i))$, where $\mathcal{L}_i$ are binary cross entropy loss for occupancy, clamped L1 loss for signed distance, and L1 losses for ESCF and CDD, respectively. For numerical stability, we set $s_i = \log(\sigma_i^2), i = \{1, 2, 3, 4\}$ as per Kendall et al. (2018). Thus, the total loss is reformulated as $\mathcal{L} = \sum_{i=1}^{4}(e^{-s^i}\mathcal{L}_i(\mathbf{W_i}) + s_i)$. During training, both the model weights $\mathbf{W_i}$ and uncertainties s_i are optimized automatically, eliminating the need for manual tuning.

A.5. Boundary-Based Occlusion Detection Methodology

This thesis introduces a robust methodology for occlusion detection based on depth discontinuities at object boundaries. The approach leverages binary segmentation masks and depth information to identify occluded regions with quantifiable confidence metrics.

The proposed occlusion detection algorithm operates on the premise that depth discontinuities at object boundaries provide reliable indicators of occlusion relationships. The procedure is formalized as follows:

1. For each object instance in the scene, we extract a binary segmentation mask $\mathbf{M}_{O_i}$ and identify its boundary contour $\partial\mathbf{M}_{O_i}$.
2. For each pixel $\mathbf{x} \in \partial\mathbf{M}_{O_i}$ on the boundary, we compute the normal vector $\mathbf{n_x}$ oriented outward from the mask.
3. We define two sampling regions along each normal vector: namely 1) internal region $R_{\text{in}}(\mathbf{x})$, and 2) external region $R_{\text{out}}(\mathbf{x})$, expressed by

$$\mathcal{R}_{\text{in}}(\mathbf{x}) = \{\mathbf{x} - b\mathbf{n_x} | 0 < b \leq \delta_{\text{in}}\}, \tag{A.1}$$

$$\mathcal{R}_{\text{out}}(\mathbf{x}) = \{\mathbf{x} + b\mathbf{n_x} | 0 < b \leq \delta_{\text{out}}\}, \tag{A.2}$$

   where δ_{in} and δ_{out} define the sampling distances inside and outside the mask, respectively.
4. We compute the average depth values within these regions:

$$d_{\text{in}}(\mathbf{x}) = \frac{1}{|\mathcal{R}_{\text{in}}(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{R}_{\text{in}}(\mathbf{x})} \mathbf{D}(\mathbf{y}) \tag{A.3}$$

$$d_{\text{out}}(\mathbf{x}) = \frac{1}{|\mathcal{R}_{\text{out}}(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{R}_{\text{out}}(\mathbf{x})} \mathbf{D}(\mathbf{y}) \tag{A.4}$$

   where $\mathbf{D}(\mathbf{y})$ represents the depth value at point $\mathbf{y}$.
5. We establish an occlusion confidence measure $C_{\text{occ}}(\mathbf{x})$ at boundary point $\mathbf{x}$:

$$C_{\text{occ}}(\mathbf{x}) = d_{\text{in}}(\mathbf{x}) - d_{\text{out}}(\mathbf{x}) \tag{A.5}$$

6. A boundary point $\mathbf{x}$ is classified as occluded if $C_{occ}(\mathbf{x}) > \xi_{\text{occ}}$, where ξ_{occ} is a threshold parameter determined empirically to account for sensor noise and sampling variations.
7. Any pixel belong to an object is classified as occluded if it is outside the binary mask and the closest boundary point to it is occluded.

As illustrated in Figure A.3, binary masks for all objects (e.g., table, kettle, cup, and human) are detected. The boundary of each mask is utilized to estimate the normal direction, which is then used to compute occlusion scores at each boundary pixel using depth information.

Pixels on the boudary highlighted in red indicate that the kettle is occluding the area outside its mask, while blue pixels show regions where the kettle is occluded by nearby objects. The rightmost column displays the occlusion boundaries for all objects.

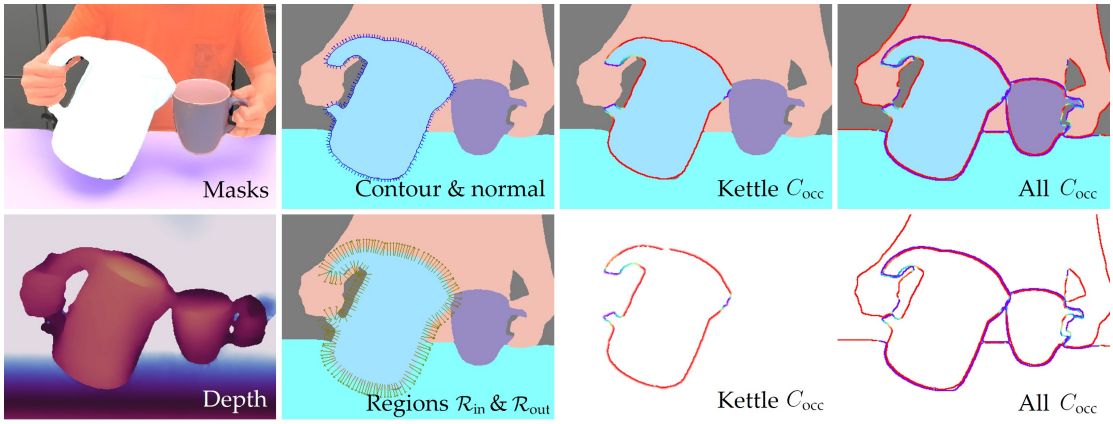


Figure A.3.: Occlusion detection.

The occlusion confidence measure provides a continuous value rather than a binary classification, allowing for probabilistic reasoning about occlusion relationships. Higher positive values of $C_{occ}(\mathbf{x})$ indicate stronger evidence of occlusion, as they correspond to scenarios where the external region is significantly closer to the camera than the internal region of the object.

This methodology offers several advantages over Spatial Tracker's internal occlusion estimation, including robustness to textural variations, independence from semantic understanding, and computational efficiency, as it operates solely on geometric properties derived from depth measurements.

APPENDIX B

Object and Spatial Relation Detection

B.1. Object Spatial Scale

Let $\mathcal{P}_i = \{\mathbf{p}_p \mid p = 1, \ldots, P_i\} \subset \mathbb{R}^3$ denote the set of candidate points on the object O_i. We formally define the *spatial scale* $\varphi_i \in \mathbb{R}$ as

$$\varphi_i = \max_{\mathbf{p},\mathbf{p}' \in \mathcal{P}_i} \|\mathbf{p} - \mathbf{p}'\|_2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm. This formulation captures the maximum distance between any pair of candidate points on O_i, thereby quantifying the overall spatial extent of the object's shape.

B.2. Contact Detection

Let $\mathcal{P}_i(t) = \{\mathbf{p}_p \mid p = 1, \ldots, P_i\}$ be the set of keypoints on object O_i at time t and $\mathcal{P}_j(t) = \{\mathbf{p}_p \mid p = 1, \ldots, P_j\}$ be the set of points on object O_j. We define the Euclidean distance between points in $\mathcal{P}_i(t)$ and $\mathcal{P}_j(t)$ as

$$d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|, \quad \forall \mathbf{p}_i \in \mathcal{P}_i(t), \quad \forall \mathbf{p}_j \in \mathcal{P}_j(t). \tag{B.1}$$

Let $\mathcal{D}_k$ be the set of the top-k smallest distances,

$$\mathcal{D}_k = \{d_{i_1 j_1}, d_{i_2 j_2}, \ldots, d_{i_k j_k}\}, \quad \text{where } d_{i_1 j_1} \leq d_{i_2 j_2} \leq \cdots \leq d_{i_k j_k}. \tag{B.2}$$

The hand and the object are considered in contact at timestep t if the mean of these top-k distances is below a threshold ξ_{contact}:

$$\frac{1}{k}\sum_{d\in D_k} d < \xi_{\text{contact}}. \tag{B.3}$$

The temporal segmentation point can be refined by also incorporating the velocities of the pair-wise relative distances. Specifically, given $v_{i_l,j_l} \in \mathbb{R}$ being the velocity of the relative distance corresponding to the l^{th} point-pair in the top-k set $\mathcal{D}_k$ and a velocity threshold that is close to zero $\xi_{\text{cont},v}$, we determine the temporal point when the average absolute value of the velocity drops below the threshold

$$\frac{1}{k}\sum_{v\in V_k} |v| < \xi_{\text{cont},v}, \quad \mathcal{V}_k = \{v_{i_1 j_1}, v_{i_2 j_2}, \ldots, v_{i_k j_k}\}. \tag{B.4}$$

The contact is detected when both conditions are satisfied.

B.3. Relative Motion Saliency

Assume P points on object O_{A}, represented as a set $\mathcal{P}_A = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \ldots, P\}$. And Q points on object O_{B}, represented as a set $\mathcal{P}_B = \{\mathbf{p}_j \in \mathbb{R}^3 \mid j = 1, \ldots, Q\}$. Each point $\mathbf{p}_i$ on O_{A} has an associated local frame $\mathcal{F}_i$, which is determined by the neighborhood of each point as explained in Section 4.1.3. The points of O_{B} are projected into each frame $\mathcal{F}_i$, yielding a transformed set of points:

$$\mathcal{P}_{i,B} = \{\tilde{\mathbf{p}}_{i,j} \in \mathbb{R}^3 \mid j = 1, \ldots, Q\}, \quad \forall i \in [1, P] \tag{B.5}$$

The velocity of each point is denoted as $\dot{\mathbf{p}}$, meaning the velocity of the transformed points in each frame is

$$\dot{\mathbf{p}}_{i,j} \quad \forall i \in [1, P], \quad j \in [1, Q]. \tag{B.6}$$

We detect the closest frames on object O_{A} to all points on object O_{B}, Define the Euclidean distances $d_{i,j}$ between each point $\mathbf{p}_i \in \mathcal{P}_A$ and each point $\mathbf{p}_j \in \mathcal{P}_B$:

$$d_{i,j} = \|\mathbf{p}_i - \mathbf{p}_j\| \tag{B.7}$$

Let D_k be the set of top-k smallest distances:

$$D_k = \{d_{i_1,j_1}, d_{i_2,j_2}, \ldots, d_{i_k,j_k}\}, \quad \text{where } d_{i_1,j_1} \leq d_{i_2,j_2} \leq \cdots \leq d_{i_k,j_k}. \tag{B.8}$$

These correspond to the top-k closest frames $F_{i_1}, F_{i_2}, \ldots, F_{i_k}$.

Mean Norm of Projected Velocities: We compute the mean of the norms of projected keypoint velocities of all points in B, across the top-k closest frames

$$\frac{1}{kQ} \sum_{i \in D_k} \sum_{j=1}^{Q} \|\dot{\mathbf{p}}_{i,j}\| \tag{B.9}$$

where the inner sum computes the sum of velocity norms over all points in B for a given frame F_i and the outer sum averages over the top-k closest frames.

B.4. Projected Relative Motion Saliency

Given object pair $(O_\mathsf{A}, O_\mathsf{B})$, and O_A is the smaller or non-hand object with spatial scale φ_A. Assume, at time t, P points on object O_A, represented as a set $\mathcal{P}_A = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \ldots, P\}$. And Q points on object O_B, represented as a set $\mathcal{P}_B = \{\mathbf{p}_j \in \mathbb{R}^3 \mid j = 1, \ldots, Q\}$.

We choose object O_B as the one providing local frames $\mathcal{F}_j$, as its broader spatial extent guarantees more precise velocity estimation. The points of O_A are projected into each frame $\mathcal{F}_j$, yielding a transformed set of points:

$$\mathcal{P}_{j,A} = \{\tilde{\mathbf{p}}_{i,j} \in \mathbb{R}^3 \mid i = 1, \ldots, P\}, \quad \forall j \in [1, Q] \tag{B.10}$$

The velocity of each point, denoted as $\dot{\mathbf{p}}$, meaning the velocity of the transformed points in each frame is

$$\dot{\mathbf{p}}_{i,j} \quad \forall i \in [1, P], \quad j \in [1, Q]. \tag{B.11}$$

We detect the top-k largest norm of velocities $v_{i,j}$

$$v_{i,j} = \|\dot{\mathbf{p}}_{i,j}\|_2, \tag{B.12}$$

and obtain

$$V_k = \{v_{i_1,j_1}, v_{i_2,j_2}, \ldots, v_{i_k,j_k}\}, \quad \text{where } v_{i_1,j_1} \geq v_{i_2,j_2} \geq \cdots \geq v_{i_k,j_k}. \tag{B.13}$$

We then compute the mean of the top-k velocities,

$$\mathsf{score}(t) = \frac{1}{k} \sum_{v \in V_k} v \tag{B.14}$$

If this score falls below the threshold $\xi_{\text{r,v}} = \hat{\xi}_{\text{r,v}} \cdot \varphi_{\mathsf{A}}$ in a period of motion, the phase is considered static; otherwise, it is categorized dynamic.

B.5. Velocity Peak Detection

Given the score value $\mathsf{score}(t)$ at each time t from Appendix B.4, we first identify dynamic phases by comparing the score with the threshold $\xi_{\text{r,v}}$. For each dynamic period, we check if $\mathsf{score}(t) > 5 \times \xi_{\text{r,v}}$. If this condition is met, a peak is detected. If not, we then evaluate the ratio of the period where $\mathsf{score}(t) > 3 \times \xi_{\text{r,v}}$. If this ratio exceeds a threshold value ξ_{peak}, a peak is also considered detected. If neither condition is satisfied, the velocity does not exhibit a significant increase during the dynamic period, and the phase is corrected to static. In this case, the relative motion is attributed to noise or minor adjustments.

B.6. Gaussian Mixture Model on Riemannian Manifolds

Recent work has demonstrated the successful application of Gaussian Mixture Models (GMMs) on Riemannian manifolds in various robotic tasks, including robot learning and control (Zeestraten, 2018; Jaquier and Calinon, 2017; Jaquier et al., 2021). A Riemannian manifold is a smooth, curved space that locally resembles Euclidean space. Many parameters in robotics naturally reside on such manifolds; for example, a unit quaternion representing orientation lies on a spherical manifold. By modeling data on Riemannian manifolds, one can leverage the intrinsic geometry, resulting in improved data efficiency and performance.

Consider a grasp pose defined by its origin $\mathbf{p}$ and orientation as a unit Quaternion $\mathbf{q}$, which lies on the manifold

$$\mathbf{z} = (\mathbf{p}^\top, \mathbf{q}^\top)^\top \in \mathcal{M} = \mathbb{R}^3 \times \mathcal{S}^3. \tag{B.15}$$

Here, the position $\mathbf{p}$ is in a Euclidean space, while the orientation $\mathbf{q}$ is on a hypersphere manifold $\mathcal{S}^3$. A GMM with K components on the manifold $\mathcal{M}$ is given by

$$p(\mathbf{p}) = \sum_{k=1}^{K} \pi_k \mathcal{N}_{\mathcal{M}}(\mathbf{p} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where each component is characterized by a mixture coefficient π_k, a mean $\boldsymbol{\mu}_k \in \mathcal{M}$, and a covariance $\boldsymbol{\Sigma}_k$ defined in the tangent space $\mathcal{T}_{\boldsymbol{\mu}_k}\mathcal{M}$.

Parameter estimation for these models is achieved using an expectation maximization (EM) algorithm, similar to the Euclidean case. However, the maximization step differs: the mean is updated using the Gauss-Newton algorithm, and the covariance is computed in the tangent space at the updated mean.

APPENDIX C

Symbols and Hyperparameters

Table C.1.: Summary of notations

Notation	Meaning
$\mathbf{A}$	an RGB image
$\mathbf{M}$	a binary mask image
$\mathbf{D}$	a depth image
$\mathbf{K}$	the camera intrinsic parameter
$\mathbf{A}_D$	an image in descriptor space
f_θ^{don}	the DON model
f_θ^{DINOv2}	the DINOv2 model
f_θ^{DIFT}	the DIFT model
f_θ^{DD}	the combination of DINOv2 and DIFT model
f_θ^{NDF}	the NDF model
f_θ^{NIFT}	the NIFT model
f_θ^{MIMO}	the MIMO model
f_θ^{pose}	the pose estimation model
SIM	cosine similarity
BM	the best matching operator
BBM	the best-buddies matching operator

Continued on next page

Table C.1 – continued from previous page

Notation	Meaning
SBBM	the soft-best-buddies matching operator
Lift	the map from 2D pixel coordinate to 3D coordinate
TPS	the Thin-Plate-Spline mapping
d	the intrinsic dimension of a principal manifold
g_1, g_2	force scaling parameters
H	the total number of candidate points
I	the number of objects
L	the number of constraints
N	the number of demonstrations
P, Q	the number of points
T	time horizon
ξ_{pixel}	soft matching threshold in image space
ξ_{dist}	point distance threshold
$\hat{\xi}_{\text{dist}}$	ratio-based distance threshold
$\xi_{\text{dist,v}}$	threshold for change rate of the distance
$\hat{\xi}_{\text{dist,v}}$	ratio-based threshold for change rate of the distance
$\xi_{\text{r,v}}$	the velocity threshold for relative motion saliency
$\hat{\xi}_{\text{r,v}}$	the ratio-based velocity threshold for relative motion saliency
$\xi_{\text{g,v}}$	velocity threshold for global motion saliency detection
ξ_{grasp}	threshold for grasp success probability
$\xi_{\text{sg,t}}$	threshold as a time window for motion segmentation algorithm
ξ_{overlap}	threshold above which segments are considered to overlap
ξ_{peak}	threshold in ratio of high velocity occupancy for peak detection
ξ_1, ξ_2	lower and upper thresholds of spatial variability
$\mathcal{H}$	the set of hands
$\mathcal{C}$	the set of constraints
$\mathcal{D}$	the set of descriptors
Θ	a set of local frame parameterization
$\mathcal{S}$	the set of canonical shapes of all objects
$\mathcal{V}$	the set of demonstration videos
$\mathcal{X}$	a set of pixels
$\mathbf{d}$	the descriptor, feature vector
$\mathbf{p}$	a position vector of a 3D point

Continued on next page

Table C.1 – continued from previous page

Notation	Meaning
$\mathbf{P}$	a 3D point cloud
$\mathbf{T}$	SE(3) transformation
$\mathbf{D}$	pose descriptor
$\mathbf{x}$	a pixel coordinate vector
C	geometric constraint
D	the dimension of the descriptor space
$\mathcal{F}$	a local frame
$\theta_{\mathcal{F}}$	parameterization of a local frame
$\mathcal{M}$	a manifold
$O, \mathcal{O}$	an object category, the set of objects
$\mathcal{P}$	the set of keypoints / candidate points
$\gamma, \mathcal{R}$	the role of the object, the set of object roles
V	a single demonstration video
φ, Φ	the spatial scale, the set of spatial scales
λ	the regularization factor of PME
κ_f	the curvature of the principal manifold
$\mathbf{f}$	a force vector
$\mathbf{h}_c$	the Coriolis and gravitational force in task space
$\mathbf{k}, \dot{\mathbf{k}}$	the position and velocity vector of a keypoint
$\boldsymbol{w}, \boldsymbol{\Sigma}_{\boldsymbol{w}}$	a weights vector of the VMP and its covariance
$\boldsymbol{\mu}_{\boldsymbol{w}}$	the mean of $\boldsymbol{w}$
$\boldsymbol{\nu}$	the explained variance
$\boldsymbol{\eta}$	the spatial variability
$\mathbb{V}$	computing variance of the input data
$\mathbf{K}$	diagonal stiffness, damping and inertia matrices
$\mathbf{S}$	the canonical shape
$\boldsymbol{\tau}$	trajectories of points
$\boldsymbol{\tau}_c$	trajectories of candidate points
$\boldsymbol{\tau}_{\mathcal{F}}$	trajectories of local frames
$\sigma(\cdot)$	the density force on the d-dimensional manifold
$\nabla\sigma(\cdot)$	the density field
$\pi_d(\cdot)$	the projection index of a principal manifold
$f_r(\cdot)$	the reconstruction function of a principal manifold
f_{vmp}	the nonlinear shape modulation of VMP
h_{vmp}	the elementary component of VMP

Continued on next page

Table C.1 – continued from previous page

Notation	Meaning
$\psi_i(\cdot)$	the squared exponential kernel
$f_r(\cdot)$	the reconstruction function of a principal manifold
$\boldsymbol{\psi}(\cdot)$	the squared exponential (SE) kernels in VMP
cont_s	static contact
prox_s	static proximity
mid_s	static middle
sep_s	static separation
grasp	grasp phase
hover	hover phase
hover_s	static hover
hover_d	dynamic hover
div	divergent motion
conv	convergent motion
snatch	snatch phase
slide	slide phase
HG	hand group
$\mathcal{G}_h$	set of hand groups
$\mathcal{S}_{\text{prox}}$	Proximity status sequence
$R_{\text{in}}(\mathbf{x})$	internal region of a pixel in the normal direction of the mask boundary
$R_{\text{out}}(\mathbf{x})$	external region of a pixel in the normal direction of the mask boundary
C_{occ}	occlusion confidence
ξ_{occ}	occlusion threshold

List of Figures

List of Tables

List of Algorithms

Bibliography

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). GPT-4 Technical Report. *arXiv:2303.08774*. *Cited on page 24.*

Ajoudani, A., Tsagarakis, N. G., Lee, J., Gabiccini, M., and Bicchi, A. (2014). Natural Redundancy Resolution in Dual-Arm Manipulation Using Configuration Dependent Stiffness (CDS) Control. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1480–1486. *Cited on page 39.*

Almeida, D. and Karayiannidis, Y. (2019). A Lyapunov-Based Approach to Exploit Asymmetries in Robotic Dual-Arm Task Resolution. In *IEEE Conference on Decision and Control (CDC)*, pages 4252–4258. *Cited on page 39.*

Amadio, F., Colome, A., and Torras, C. (2019). Exploiting Symmetries in Reinforcement Learning of Bimanual Robotic Tasks. *IEEE Robotics and Automation Letters*, 4(2):1838–1845. *Cited on page 39.*

Amir, S., Gandelsman, Y., Bagon, S., and Dekel, T. (2022). Deep ViT Features as Dense Visual Descriptors. In *Euro. Conf. on Computer Vision Workshops (ECCVW)*. *Cited on page 15.*

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A Survey of Robot Learning from Demonstration. *Robotics and Autonomous Systems*, 57(5):469–483. *Cited on page 11.*

Asfour, T., Wächter, M., Kaul, L., Rader, S., Weiner, P., Ottenhaus, S., Grimm, R., Zhou, Y., Grotz, M., and Paus, F. (2019). ARMAR-6: A High-Performance Humanoid for Human-Robot Collaboration in Real World Scenarios. *IEEE Robotics and Automation Magazine*, 26(4):108–121. *Cited on page 151.*

Ausserlechner, P., Haberger, D., Thalhammer, S., Weibel, J.-B., and Vincze, M. (2024). ZS6D: Zero-shot 6D Object Pose Estimation using Vision Transformers. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 463–469. *Cited on page 72.*

Bai, J. and Perron, P. (2003). Computation and Analysis of Multiple Structural Change Models. *Journal of Applied Econometrics*, 18(1):1–22. *Cited on page 45.*

Bandura, A. (1977). *Social Learning Theory*, volume 1. Englewood cliffs Prentice Hall. *Cited on page 1.*

Barbič, J., Safonova, A., Pan, J.-Y., Faloutsos, C., Hodgins, J. K., and Pollard, N. S. (2004). Segmenting Motion Capture Data into Distinct Behaviors. In *Graphics Interface*, volume 62, pages 185–194. *Cited on pages 43 and 48.*

Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural computation*, 15(6):1373–1396. *Cited on page 28.*

Billard, A. G., Calinon, S., and Dillmann, R. (2016). Learning from Humans. In *Springer Handbook of Robotics*, Springer Handbooks, pages 1995–2014. Springer. *Cited on pages 11, 187, and 193.*

Biza, O., Thompson, S., Pagidi, K. R., Kumar, A., van der Pol, E., Walters, R., Kipf, T., van de Meent, J.-W., Wong, L. L. S., and Platt, R. (2023). One-shot Imitation Learning via Interaction Warping. In *Conference on Robot Learning (CoRL)*, volume 229, pages 2519–2536. *Cited on pages 23 and 31.*

Bizzi, E., Mussa-Ivaldi, F. A., and Giszter, S. (1991). Computations Underlying the Execution of Movement: A Biological Perspective. *Science*, 253(5017):287–291. *Cited on page 41.*

Budninskiy, M., Yin, G., Feng, L., Tong, Y., and Desbrun, M. (2019). Parallel Transport Unfolding: A Connection-Based Manifold Learning Approach. *SIAM Journal on Applied Algebra and Geometry*, 3(2):266–291. *Cited on page 28.*

Burke, C. J., Tobler, P. N., Baddeley, M., and Schultz, W. (2010). Neural Mechanisms of Observational Learning. 107(32):14431–14436. *Cited on page 1.*

Cai, Y., Gao, J., Pohl, C., and Asfour, T. (2024). Visual Imitation Learning of Task-Oriented Object Grasping and Rearrangement. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 364–371. *Cited on pages 60, 68, 79, 80, 156, and 183.*

Calinon, S. (2016). A Tutorial on Task-Parameterized Movement Learning and Retrieval. *Intelligent Service Robotics*, 9(1):1–29. *Cited on pages 19, 34, and 50.*

Calinon, S. (2018). Learning from Demonstration (Programming by Demonstration). In *Encyclopedia of Robotics*, pages 1–8. Springer Berlin Heidelberg. *Cited on page 11.*

Calinon, S., Bruno, D., and Caldwell, D. G. (2014). A Task-Parameterized Probabilistic Model with Minimal Intervention Control. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3339–3344. *Cited on pages 18, 19, and 34.*

Calinon, S., D'halluin, F., Sauser, E., Caldwell, D., and Billard, A. (2010). Learning and Reproduction of Gestures by Imitation. *IEEE Robotics & Automation Magazine*, 17(2):44–54. *Cited on page 43.*

Chang, W.-D., Hogan, F., Fujimoto, S., Meger, D., and Dudek, G. (2025). Generalizable Imitation Learning Through Pre-Trained Representations. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1–8. *Cited on pages 21, 22, and 31.*

Chen, H., Niu, Y., Hong, K., Liu, S., Wang, Y., Li, Y., and Driggs-Campbell, K. (2023). Predicting Object Interactions with Behavior Primitives: An Application in Stowing Tasks. In *Conference on Robot Learning (CoRL)*, volume 229, pages 358–373. *Cited on pages 23 and 30.*

Chen, Y., Wu, T., Wang, S., Feng, X., Jiang, J., McAleer, S. M., Dong, H., Lu, Z., Zhu, S.-C., and Yang, Y. (2022). Towards Human-Level Bimanual Dexterous Manipulation with Reinforcement Learning. In *Neural Information Processing Systems (NeurIPS)*, volume 35, pages 5150–5163. Curran Associates, Inc. *Cited on page 39.*

Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., and Yang, Y. (2023). Segment and Track Anything. *arXiv:2305.06558*. *Cited on page 86.*

Choy, C. B., Gwak, J., Savarese, S., and Chandraker, M. (2016). Universal Correspondence Network. In *Neural Information Processing Systems (NeurIPS)*, pages 2406–2414. *Cited on page 54.*

Chun, E., Du, Y., Simeonov, A., Lozano-Perez, T., and Kaelbling, L. (2023). Local Neural Descriptor Fields: Locally Conditioned Object Representations for Manipulation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1830–1836. *Cited on pages 23, 30, and 34.*

Cunningham, E., Cobb, A., and Jha, S. (2022). Principal Manifold Flows. *arXiv:2202.07037*. *Cited on page 28.*

Damiani, M. L., Hachem, F., Issa, H., Ranc, N., Moorcroft, P., and Cagnacci, F. (2018). Cluster-Based Trajectory Segmentation with Local Noise. *Data Mining and Knowledge Discovery*, 32(4):1017–1055. *Cited on page 44.*

Das, N., Bechtle, S., Davchev, T., Jayaraman, D., Rai, A., and Meier, F. (2021). Model-Based Inverse Reinforcement Learning from Visual Demonstrations. In *Conference on Robot Learning (CoRL)*, volume 155, pages 1930–1942. *Cited on page 32.*

Dekel, T., Oron, S., Rubinstein, M., Avidan, S., and Freeman, W. T. (2015). Best-Buddies Similarity for Robust Template Matching. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2021–2029. *Cited on page 74.*

DeLoache, J. S., Uttal, D. H., and Rosengren, K. S. (2004). Scale Errors Offer Evidence for a Perception-Action Dissociation Early in Life. *Science*, 304(5673):1027–1029. *Cited on pages 2 and 13.*

Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. J. (2021). Vector Neurons: A General Framework for SO(3)-Equivariant Networks. In *Intl. Conf. on Computer Vision (ICCV)*, pages 12180–12189. *Cited on pages 57 and 78.*

Despinoy, F., Bouget, D., Forestier, G., Penet, C., Zemiti, N., Poignet, P., and Jannin, P. (2016). Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training. *IEEE Transactions on Bio-Medical Engineering*, 63(6):1280–1291. *Cited on page 43.*

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2018). SuperPoint: Self-Supervised Interest Point Detection and Description. In *CVPR Deep Learning for Visual SLAM Workshop*. *Cited on page 14.*

Di Palo, N. and Johns, E. (2024a). DINOBot: Robot Manipulation via Retrieval and Alignment with Vision Foundation Models. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2798–2805. *Cited on pages 15 and 19.*

Di Palo, N. and Johns, E. (2024b). Keypoint Action Tokens Enable In-Context Imitation Learning in Robotics. In *Robotics: Science and Systems (R:SS)*. *Cited on pages 15, 19, 21, 22, and 31.*

Do, T.-T., Nguyen, A., and Reid, I. (2018). AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1–5. *Cited on pages 128 and 191.*

Dodds, Z., Jägersand, M., Hager, G., and Toyama, K. (1999). *A Hierarchical Vision Architecture for Robotic Manipulation Tasks*, volume 1542, pages 312–330. Springer Berlin Heidelberg. *Cited on pages 26, 28, and 30.*

Dong, Z., Li, Z., Yan, Y., Calinon, S., and Chen, F. (2022). Passive Bimanual Skills Learning From Demonstration With Motion Graph Attention Networks. *IEEE Robotics and Automation Letters*, 7(2):4917–4923. *Cited on page 39.*

Dreher, C. R. G. and Asfour, T. (2022). Learning Temporal Task Models from Human Bimanual Demonstrations. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 7664–7671. *Cited on page 185.*

Dreher, C. R. G., Wächter, M., and Asfour, T. (2020a). Learning Object-Action Relations from Bimanual Human Demonstration Using Graph Networks. *IEEE Robotics and Automation Letters*, 5(1):187–194. *Cited on page 40.*

Dreher, C. R. G., Wachter, M., and Asfour, T. (2020b). Learning Object-Action Relations from Bimanual Human Demonstration Using Graph Networks. *IEEE Robotics and Automation Letters*, 5(1):187–194. *Cited on pages 42, 46, 47, 49, 50, 158, and 172.*

Duchon, J. (1977). Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces. In *Constructive Theory of Functions of Several Variables*, volume 571, pages 85–100. Springer Berlin Heidelberg. *Cited on pages 16 and 76.*

Dümbgen, L. (1998). On Tyler's M-functional of scatter in high dimension. *Annals of the Institute of Statistical Mathematics*, 50:471–491. *Cited on page 136.*

Dwibedi, D., Tompson, J., Lynch, C., and Sermanet, P. (2018). Learning Actionable Representations from Visual Observations. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1577–1584. *Cited on pages 18 and 36.*

Einbeck, J. (1989). Principal Curves and Surfaces: Data Visualization, Compression, and Beyond. page 47. *Cited on page 28.*

Elhamifar, E. and Vidal, R. (2009). Sparse Subspace Clustering. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2797. *Cited on page 43.*

Firoozi, R., Tucker, J., Tian, S., Majumdar, A., Sun, J., Liu, W., Zhu, Y., Song, S., Kapoor, A., Hausman, K., Ichter, B., Driess, D., Wu, J., Lu, C., and Schwager, M. (2025). Foundation Models in Robotics: Applications, Challenges, and the Future. *Intl. Journal of Robotics Research*, 44(5):701–739. *Cited on page 15.*

Flash, T. and Hochner, B. (2005). Motor Primitives in Vertebrates and Invertebrates. *Current Opinion in Neurobiology*, 15(6):660–666. *Cited on page 41.*

Florence, P., Manuelli, L., and Tedrake, R. (2020). Self-Supervised Correspondence in Visuomotor Policy Learning. *IEEE Robotics and Automation Letters*, 5(2):492–499. *Cited on pages 15, 19, and 55.*

Florence, P. R., Manuelli, L., and Tedrake, R. (2018). Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, volume 87, pages 373–385. *Cited on pages 14, 15, 27, 30, 54, 55, and 183.*

Fod, A. (2002). Automated Derivation of Primitives for Movement Classification. *Autonomous Robots*, 12(1):39–54. *Cited on pages 43, 44, 46, 47, and 159.*

Franzese, G., Celemin, C., and Kober, J. (2020). Learning Interactively to Resolve Ambiguity in Reference Frame Selection. In *Conference on Robot Learning (CoRL)*, volume 155, pages 1298–1311. *Cited on page 35.*

Franzese, G., Rosa, L. d. S., Verburg, T., Peternel, L., and Kober, J. (2024). Interactive Imitation Learning of Bimanual Movement Primitives. *IEEE/ASME Transactions on Mechatronics*, 29(5):4006–4018. *Cited on page 39.*

Fu, Z., Zhao, Q., Wu, Q., Wetzstein, G., and Finn, C. (2024a). HumanPlus: Humanoid Shadowing and Imitation from Humans. In *Conference on Robot Learning (CoRL). Cited on page 12.*

Fu, Z., Zhao, T. Z., and Finn, C. (2024b). Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. In *Conference on Robot Learning (CoRL). Cited on pages 39 and 153.*

Ganapathi, A., Sundaresan, P., Thananjeyan, B., Balakrishna, A., Seita, D., Grannen, J., Hwang, M., Hoque, R., Gonzalez, J. E., Jamali, N., Yamane, K., Iba, S., and Goldberg, K. (2021). Learning Dense Visual Correspondences in

Simulation to Smooth and Fold Real Fabrics. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 11515–11522. *Cited on pages 15, 27, and 30.*

Gao, J., Jin, X., Krebs, F., Jaquier, N., and Asfour, T. (2024). Bi-KVIL: Keypoints-based Visual Imitation Learning of Bimanual Manipulation Tasks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 16850–16857. *Cited on pages 130, 131, 145, 149, 151, and 152.*

Gao, J., Tao, Z., Jaquier, N., and Asfour, T. (2023). K-VIL: Keypoints-Based Visual Imitation Learning. *IEEE Trans. on Robotics*, 39(5):3888–3908. *Cited on pages 68, 91, 94, 96, 97, 98, 99, 105, 106, 107, 108, 109, 110, 112, 114, 115, 116, 117, 119, 120, 122, and 124.*

Gao, J., Zhou, Y., and Asfour, T. (2018). Projected Force-Admittance Control for Compliant Bimanual Tasks. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 1–9. *Cited on page 39.*

Gao, W. and Tedrake, R. (2021a). kPAM 2.0: Feedback Control for Category-Level Robotic Manipulation. *IEEE Robotics and Automation Letters*, 6(2):2962–2969. *Cited on pages 26 and 30.*

Gao, W. and Tedrake, R. (2021b). kPAM-SC: Generalizable Manipulation Planning using KeyPoint Affordance and Shape Completion. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6527–6533. *Cited on pages 26 and 30.*

Gergely, G., Bekkering, H., and Király, I. (2002). Rational Imitation in Preverbal Infants. *Nature*, 415(6873):755–755. *Cited on page 31.*

Graybiel, A. M. (1998). The Basal Ganglia and Chunking of Action Repertoires. *Neurobiology of Learning and Memory*, 70(1-2):119–136. *Cited on pages 3, 40, and 41.*

Gribovskaya, E. and Billard, A. (2008). Combining Dynamical Systems Control and Programming by Demonstration for Teaching Discrete Bimanual Coordination Tasks to a Humanoid Robot. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 33–40. *Cited on page 38.*

Gridseth, M., Ramirez, O., Quintero, C. P., and Jagersand, M. (2016). ViTa: Visual Task Specification Interface for Manipulation with Uncalibrated Visual Servoing. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3434–3440. *Cited on pages 26, 28, and 30.*

Guan, J., Hao, Y., Wu, Q., Li, S., and Fang, Y. (2024). A Survey of 6DoF Object Pose Estimation Methods for Different Application Scenarios. *Sensors*, 24(4):1076. *Cited on page 19.*

Guiard, Y. (1987a). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*, 19(4):486–517. *Cited on pages 34 and 38.*

Guiard, Y. (1987b). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behaviour*, 19(4):486–517. *Cited on page 38.*

Gutzeit, L. (2022). Hierarchical Segmentation of Human Manipulation Movements. In *International Conference on Pattern Recognition (ICPR)*, pages 2742–2748. *Cited on pages 44, 45, and 50.*

Gutzeit, L. and Kirchner, F. (2022). Unsupervised Segmentation of Human Manipulation Movements Into Building Blocks. *IEEE Access*, 10:125723–125734. *Cited on pages 40, 44, 45, 47, 48, 49, and 164.*

Gweon, H. and Schulz, L. (2011). 16-Month-Olds Rationally Infer Causes of Failed Actions. *Science*, 332(6037):1524–1524. *Cited on pages 1 and 31.*

Gweon, H., Tenenbaum, J. B., and Schulz, L. E. (2010). Infants Consider Both the Sample and the Sampling Process in Inductive Generalization. *Proceedings of the National Academy of Sciences*, 107(20):9066–9071. *Cited on pages 1, 2, and 192.*

Hachem, F. and Damiani, M. L. (2018). Periodic Stops Discovery through Density-Based Trajectory Segmentation. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 584–587. *Cited on page 44.*

Hadjivelichkov, D. and Kanoulas, D. (2022). Fully Self-Supervised Class Awareness in Dense Object Descriptors. In *Conference on Robot Learning (CoRL)*, volume 164, pages 1522–1531. *Cited on pages 15 and 30.*

Hadjivelichkov, D., Zwane, S., Deisenroth, M. P., Agapito, L., and Kanoulas, D. (2022). One-Shot Transfer of Affordance Regions? AffCorrs! In *Conference on Robot Learning (CoRL)*, pages 550–560. *Cited on pages 128, 185, and 191.*

Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, 53(2):217–288. *Cited on page 90.*

Hastie, T. (1984). Principal Curves and Surfaces. Technical Report LCS-11, Stanford University. *Cited on page 28.*

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. *Cited on page 55.*

Hespanha, J. P., Dodds, Z., Hager, G. D., and Morse, A. S. (1999). What Tasks Can Be Performed with an Uncalibrated Stereo Vision System? *Intl. Journal on Computer Vision*, 35:65–85. *Cited on pages 26, 28, and 30.*

Huang, H., Lin, F., Hu, Y., Wang, S., and Gao, Y. (2024a). CoPa: General Robotic Manipulation through Spatial Constraints of Parts with Foundation Models. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 9488–9495. *Cited on pages 25 and 31.*

Huang, W., Wang, C., Li, Y., Zhang, R., and Fei-Fei, L. (2024b). ReKep: Spatio-Temporal Reasoning of Relational Keypoint Constraints for Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, volume 270, pages 4573–4602. *Cited on pages 15, 16, 19, 21, 24, 25, 31, and 192.*

Huang, Y., Silvério, J., Rozo, L., and Caldwell, D. G. (2018). Generalized Task-Parameterized Skill Learning. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5667–5474. *Cited on pages 19 and 34.*

Huang, Z., Xu, J., Dai, S., Xu, K., Zhang, H., Huang, H., and Hu, R. (2023). NIFT: Neural Interaction Field and Template for Object Manipulation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1875–1881. *Cited on pages 17, 23, 30, 34, 54, 57, 58, 59, and 183.*

Ichnowski, J., Kerr, J., Avigal, Y., and Goldberg, K. (2021). Dex-NeRF: Using a Neural Radiance Field to Grasp Transparent Objects. In *Conference on Robot Learning (CoRL)*, volume 164, pages 526–536. *Cited on page 56.*

Jaquier, N. and Calinon, S. (2017). Gaussian Mixture Regression on Symmetric Positive Definite Matrices Manifolds: Application to Wrist Motion Estimation with sEMG. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 59–64. *Cited on page 206.*

Jaquier, N., Rozo, L., Caldwell, D. G., and Calinon, S. (2021). Geometry-aware Manipulability Learning, Tracking, and Transfer. *Intl. Journal of Robotics Research*, 40(2-3):624–650. *Cited on page 206.*

Jiang, T., Lu, P., Zhang, L., Ma, N., Han, R., Lyu, C., Li, Y., and Chen, K. (2023). RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose. *arXiv:2303.07399*. *Cited on page 88.*

Jiang, Z., Zhu, Y., Svetlik, M., Fang, K., and Zhu, Y. (2021). Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations. In *Robotics: Science and Systems (R:SS)*. *Cited on pages 128 and 191.*

Jin, J. and Jagersand, M. (2022). Generalizable Task Representation Learning from Human Demonstration Videos: A Geometric Approach. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2504–2510. *Cited on pages 12, 21, 27, 29, 31, 32, 33, 65, and 126.*

Jin, J., Petrich, L., Dehghan, M., and Jagersand, M. (2020a). A Geometric Perspective on Visual Imitation Learning. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5194–5200. *Cited on pages 27, 31, and 32.*

Jin, J., Petrich, L., Zhang, Z., Dehghan, M., and Jagersand, M. (2020b). Visual Geometric Skill Inference by Watching Human Demonstration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 8985–8991. *Cited on pages 27, 31, and 32.*

Jonnavittula, A., Parekh, S., and P Losey, D. (2025). VIEW: Visual Imitation Learning with Waypoints. *Autonomous Robots*, 49(1):1–26. *Cited on page 19.*

Ju, Y., Hu, K., Zhang, G., Zhang, G., Jiang, M., and Xu, H. (2024). Robo-ABC: Affordance Generalization beyond Categories via Semantic Correspondence for Robot Manipulation. In *Euro. Conf. on Computer Vision (ECCV)*, pages 222–239. *Cited on pages 15 and 19.*

Karnan, H., Torabi, F., Warnell, G., and Stone, P. (2022a). Adversarial Imitation Learning from Video Using a State Observer. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2452–2458. *Cited on pages 18, 36, and 126.*

Karnan, H., Warnell, G., Xiao, X., and Stone, P. (2022b). VOILA: Visual-Observation-Only Imitation Learning for Autonomous Navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2497–2503. *Cited on pages 12, 14, 18, and 30.*

Kartmann, R., Liu, D., and Asfour, T. (2021). Semantic Scene Manipulation Based on 3D Spatial Object Relations and Language Instructions. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 306–313. *Cited on pages 128 and 191.*

Kartmann, R., Zhou, Y., Liu, D., Paus, F., and Asfour, T. (2020). Representing Spatial Object Relations as Parametric Polar Distribution for Scene Manipulation Based on Verbal Commands. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 8373–8380. *Cited on page 185.*

Karunratanakul, K., Yang, J., Zhang, Y., Black, M. J., Muandet, K., and Tang, S. (2020). Grasping Field: Learning Implicit Representations for Human Grasps. In *International Conference on 3D Vision (3DV)*, pages 333–344. *Cited on page 57.*

Kataoka, S., Ghasemipour, S. K. S., Freeman, D., and Mordatch, I. (2022). Bi-Manual Manipulation and Attachment via Sim-to-Real Reinforcement Learning. *arXiv:2203.08277*. *Cited on page 39.*

Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. (2003). Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In *Symposium on geometry processing*, volume 6, pages 156–164. *Cited on page 59.*

Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491. *Cited on page 199.*

Kerr, J., Kim, C. M., Goldberg, K., Kanazawa, A., and Tancik, M. (2023). LERF: Language Embedded Radiance Fields. In *Intl. Conf. on Computer Vision (ICCV)*, pages 19672–19682. *Cited on page 16.*

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal Detection of Changepoints with a Linear Computational Cost. *Journal of the American Statistical Association*, 107(500):1590–1598. *Cited on page 45.*

Kim, H., Ohmura, Y., and Kuniyoshi, Y. (2021). Transformer-Based Deep Imitation Learning for Dual-Arm Robot Manipulation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 8965–8972. *Cited on pages 39 and 153.*

Kim, H., Ohmura, Y., and Kuniyoshi, Y. (2024). Goal-Conditioned Dual-Action Imitation Learning for Dexterous Dual-Arm Robot Manipulation. *IEEE Trans. on Robotics*, 40:2287–2305. *Cited on pages 39, 40, and 153.*

Kimmerle, M., Ferre, C. L., Kotwica, K. A., and Michel, G. F. (2010). Development of Role-Differentiated Bimanual Manipulation during the Infant's First Year. *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology*, 52:168–180. *Cited on pages 34 and 38.*

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment Anything. In *Intl. Conf. on Computer Vision (ICCV)*, pages 3992–4003. *Cited on pages 15, 70, and 84.*

Knaust, M. and Koert, D. (2021). Guided Robot Skill Learning: A User-Study on Learning Probabilistic Movement Primitives with Non-Experts. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 514–521. *Cited on page 39.*

Ko, P.-C., Mao, J., Du, Y., Sun, S.-H., and Tenenbaum, J. B. (2024). Learning to Act from Actionless Videos through Dense Correspondences. In *Intl. Conf. on Learning Representations (ICLR)*. *Cited on page 36.*

Kober, J., Gienger, M., and Steil, J. J. (2015). Learning Movement Primitives for Force Interaction Tasks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3192–3199. *Cited on page 35.*

Krebs, F. and Asfour, T. (2022). A Bimanual Manipulation Taxonomy. *IEEE Robotics and Automation Letters*, 7(4):11031–11038. *Cited on pages 5, 38, 40, 126, 129, 130, 135, 139, 140, 141, 147, and 153.*

Krishnan, S., Garg, A., Patil, S., Lea, C., Hager, G., Abbeel, P., and Goldberg, K. (2017). Transition State Clustering: Unsupervised Surgical Trajectory Segmentation for Robot Learning. *International Journal of Robotics Research*, 36(13-14):1595–1618. *Cited on pages 43 and 45.*

Kroemer, O., Niekum, S., and Konidaris, G. (2021). A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms. *Journal of machine learning research*, 22(30):1–82. *Cited on pages 4, 11, 12, 14, 33, and 155.*

Kulić, D., Ott, C., Lee, D., Ishikawa, J., and Nakamura, Y. (2012). Incremental Learning of Full Body Motion Primitives and Their Sequencing through Human Motion Observation. *Intl. Journal of Robotics Research*, 31(3):330–345. *Cited on page 45.*

Lee, J. and Chang, P. H. (2015). Redundancy Resolution for Dual-Arm Robots Inspired by Human Asymmetric Bimanual Action: Formulation and Experiments. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6058–6065. *Cited on page 39.*

Lee, S. H., Suh, I. H., Calinon, S., and Johansson, R. (2015). Autonomous Framework for Segmenting Robot Trajectories of Manipulation Task. *Autonomous Robots*, 38(2):107–141. *Cited on page 43.*

Lin, J., Zeng, A., Wang, H., Zhang, L., and Li, Y. (2023). One-Stage 3D Whole-Body Mesh Recovery With Component Aware Transformer. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 21159–21168. *Cited on page 88.*

Lin, X., So, J., Mahalingam, S., Liu, F., and Abbeel, P. (2024). SpawnNet: Learning Generalizable Visuomotor Skills from Pre-trained Network. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4781–4787. *Cited on pages 15 and 22.*

Lioutikov, R., Neumann, G., Maeda, G., and Peters, J. (2015). Probabilistic Segmentation Applied to an Assembly Task. In *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pages 533–540. *Cited on pages 44, 45, 48, and 49.*

Lioutikov, R., Neumann, G., Maeda, G., and Peters, J. (2017). Learning Movement Primitive Libraries through Probabilistic Segmentation. *Intl. Journal of Robotics Research*, 36(8):879–894. *Cited on pages 44, 48, and 49.*

Lipson, L., Teed, Z., and Deng, J. (2021). RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. In *International Conference on 3D Vision (3DV)*, pages 218–227. *Cited on page 197.*

Liu, J., Chen, Y., Dong, Z., Wang, S., Calinon, S., Li, M., and Chen, F. (2022). Robot Cooking with Stir-fry: Bimanual Non-prehensile Manipulation of Semi-fluid Objects. *IEEE Robotics and Automation Letters*, 7(2):5159–5166. *Cited on pages 38, 39, 40, and 153.*

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., and Zhang, L. (2024). Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection. In *Euro. Conf. on Computer Vision (ECCV)*, volume 15105, pages 38–55. *Cited on pages 70 and 84.*

Liu, Y., Gupta, A., Abbeel, P., and Levine, S. (2018). Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1118–1125. *Cited on pages 11, 12, 18, 21, 35, and 36.*

Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1150–1157. *Cited on page 14.*

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., et al. (2019). MediaPipe: A Framework for Building Perception Pipelines. *arXiv:1906.08172*. *Cited on page 87.*

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., and State, G. (2021). Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*. *Cited on page 179.*

Manuelli, L., Gao, W., Florence, P., and Tedrake, R. (2019). kPAM: KeyPoint Affordances for Category-Level Robotic Manipulation. In *ISRR*, volume 20 of *Springer Proceedings in Advanced Robotics*, pages 132–157. *Cited on pages 20, 26, 30, and 35.*

Manuelli, L., Li, Y., Florence, P., and Tedrake, R. (2020). Keypoints into the Future: Self-Supervised Correspondence in Model-Based Reinforcement Learning. In *Conference on Robot Learning (CoRL)*, volume 155, pages 693–710. *Cited on pages 15, 30, and 55.*

Maxim, A., Lazar, C., Burlacu, A., and Copot, C. (2012). Robotic visual servoing system based on SIFT features. In *International Conference on System Theory, Control and Computing (ICSTCC)*, pages 1–6. *Cited on page 14.*

Meixner, A., Krebs, F., Jaquier, N., and Asfour, T. (2023). An Evaluation of Action Segmentation Algorithms on Bimanual Manipulation Datasets. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4912–4919. *Cited on pages 42, 43, 47, 48, and 158.*

Meltzoff, A. N. (1995). Understanding the Intentions of Others: Re-enactment of Intended Acts by 18-Month-Old Children. *Developmental Psychology*, 31:838–850. *Cited on pages 1, 6, and 31.*

Meltzoff, A. N. and Prinz, W. (2002). *The Imitative Mind: Development, Evolution and Brain Bases*, volume 6. Cambridge University Press. *Cited on page 1.*

Meltzoff, A. N., Waismeyer, A., and Gopnik, A. (2012). Learning about Causes from People: Observational Causal Learning in 24-Month-Old Infants. *Developmental Psychology*, 48(5):1215–1228. *Cited on pages 1, 2, and 6.*

Mendez, A., Prados, A., Menendez, E., and Barber, R. (2024). Everyday Objects Rearrangement in a Human-like Manner via Robotic Imagination and Learning from Demonstration. *IEEE Access*, 12:92098–92119. *Cited on page 51.*

Mendonca, R., Bahl, S., and Pathak, D. (2023). Structured World Models from Human Videos. In *Robotics: Science and Systems (R:SS)*. *Cited on page 31.*

Meng, K. and Eloyan, A. (2021). Principal Manifold Estimation via Model Complexity Selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):369–394. *Cited on pages 28, 53, 63, and 68.*

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy Networks: Learning 3D Reconstruction in Function Space. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470. *Cited on pages 57, 58, and 78.*

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2021). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Communications of the ACM*, 65(1):99–106. *Cited on page 56.*

Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K., and Lee, H. (2020). Unsupervised Learning of Object Structure and Dynamics from Videos. In *Neural Information Processing Systems (NeurIPS)*, pages 92–102. *Cited on page 32.*

Mirrazavi Salehian, S. S., Figueroa, N., and Billard, A. (2018). A Unified Framework for Coordinated Multi-Arm Motion Planning. *The International Journal of Robotics Research*, 37(10):1205–1232. *Cited on page 39.*

Moon, G. (2023). Bringing Inputs to Shared Domains for 3D Interacting Hands Recovery in the Wild. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 17028–17037. *Cited on page 88.*

Muhlig, M., Gienger, M., Hellbach, S., Steil, J. J., and Goerick, C. (2009a). Task-Level Imitation Learning Using Variance-Based Movement Optimization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1177–1184. *Cited on pages 5 and 130.*

Muhlig, M., Gienger, M., Steil, J. J., and Goerick, C. (2009b). Automatic Selection of Task Spaces for Imitation Learning. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4996–5002. *Cited on pages 5, 35, 89, and 130.*

Müller, T., Evans, A., Schied, C., and Keller, A. (2022). Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15. *Cited on pages 15 and 56.*

Mussa–Ivaldi, F. A. and Bizzi, E. (2000). Motor Learning through the Combination of Primitives. *Philosophical Transactions of the Royal Society of London*, 355(1404):1755–1769. *Cited on page 41.*

Niekum, S., Osentoski, S., Konidaris, G., and Barto, A. G. (2012). Learning and Generalization of Complex Tasks from Unstructured Demonstrations. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5239–5246. *Cited on pages 35, 43, and 45.*

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2024). DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*. *Cited on pages 15, 16, 21, 54, 60, and 69.*

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., and Peters, J. (2018). An Algorithmic Perspective on Imitation Learning. *Foundations and Trends in Robotics*, 7(1-2):1–179. *Cited on pages 11 and 64.*

Pairet, È., Ardón, P., Mistry, M., and Petillot, Y. (2019). Learning and Composing Primitive Skills for Dual-arm Manipulation. In *Towards Autonomous Robotic Systems: 20th Annual Conference*, volume 11649, pages 65–77. *Cited on page 39.*

Paraschos, A., Daniel, C., Peters, J., and Neumann, G. (2018). Using Probabilistic Movement Primitives in Robotics. *Autonomous Robots*, 42(3):529–551. *Cited on page 65.*

Pari, J., Muhammad, N., Pandian, S., and Pinto, L. (2022). The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems (R:SS)*, page 13. *Cited on pages 12, 19, and 118.*

Park, H. A. and Lee, C. S. G. (2015). Extended Cooperative Task Space for Manipulation Tasks of Humanoid Robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6088–6093. *Cited on page 39.*

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174. *Cited on pages 57 and 78.*

Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and Generalization of Motor Skills by Learning from Demonstration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 763–768. *Cited on pages 3 and 40.*

Patel, A., Wang, A., Radosavovic, I., and Malik, J. (2022). Learning to Imitate Object Interactions from Internet Videos. *arXiv:2211.13225*. *Cited on page 20.*

Pathak, D., Mahmoudieh, P., Luo, G., Agrawal, P., Chen, D., Shentu, F., Shelhamer, E., Malik, J., Efros, A. A., and Darrell, T. (2018). Zero-Shot Visual Imitation. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2050–2053. *Cited on pages 19 and 126.*

Pearson, K. (1901). LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572. *Cited on pages 28 and 68.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. *Cited on page 102.*

Perez-D'Arpino, C. and Shah, J. A. (2017). C-LEARN: Learning Geometric Constraints from Demonstrations for Multi-Step Manipulation in Shared Autonomy. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4058–4065. *Cited on page 19.*

Pfrommer, S., Halm, M., and Posa, M. (2021). Contactnets: Learning Discontinuous Contact Dynamics with Smooth, Implicit Representations. In *Conference on Robot Learning (CoRL)*, pages 2279–2291. *Cited on page 57.*

Prados, A., Espinoza, G., Moreno, L., and Barber, R. (2025). Segment, Compare, and Learn: Creating Movement Libraries of Complex Task for Learning from Demonstration. *Biomimetics*, 10(1):64. *Cited on pages 40, 43, 44, 45, 46, 47, 48, 49, and 164.*

Press, W. H. and Teukolsky, S. A. (1990). Savitzky-Golay Smoothing Filters. *Computers in Physics*, 4(6):669–672. *Cited on page 87.*

Prokudin, S., Lassner, C., and Romero, J. (2019). Efficient Learning on Point Clouds with Basis Point Sets. In *Intl. Conf. on Computer Vision (ICCV)*, pages 4332–4341. *Cited on pages 23 and 58.*

Qin, Y., Wu, Y.-H., Liu, S., Jiang, H., Yang, R., Fu, Y., and Wang, X. (2022). DexMV: Imitation Learning for Dexterous Manipulation from Human Videos. In *Euro. Conf. on Computer Vision (ECCV)*, pages 570–587. Springer Nature Switzerland. *Cited on page 12.*

Qin, Z., Fang, K., Zhu, Y., Fei-Fei, L., and Savarese, S. (2020). KETO: Learning Keypoint Representations for Tool Manipulation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 7278–7285. *Cited on page 32.*

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning Transferable Visual Models from Natural Language Supervision. In *Intl. Conf. on Machine Learning (ICML)*, pages 8748–8763. *Cited on pages 15 and 62.*

Ramachandruni, K., Babu V., M., Majumder, A., Dutta, S., and Kumar, S. (2020). Attentive Task-Net: Self Supervised Task-Attention Network for Imitation Learning using Video Demonstration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4760–4766. *Cited on page 12.*

Ranzinger, M., Heinrich, G., Kautz, J., and Molchanov, P. (2024). AM-RADIO: Agglomerative Vision Foundation Model – Reduce All Domains Into One. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12490–12500. *Cited on pages 15 and 62.*

Rawlings, B. and Legare, C. H. (2021). Toddlers, Tools, and Tech: The Cognitive Ontogenesis of Innovation. *Trends in Cognitive Sciences*, 25(1):81–92. *Cited on page 13.*

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685. *Cited on page 15.*

Romero, J., Tzionas, D., and Black, M. J. (2017). Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics*, 36(6):1–17. *Cited on pages 88, 89, 134, and 198.*

Roweis, S. T. and Saul, L. K. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *science*, 290(5500):2323–2326. *Cited on page 28.*

Rozo, L., Guo, M., Kupcsik, A. G., Todescato, M., Schillinger, P., Giftthaler, M., Ochs, M., Spies, M., Waniek, N., Kesper, P., and Büerger, M. (2020). Learning and Sequencing of Object-Centric Manipulation Skills for Industrial Tasks. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 9072–9079. *Cited on pages 45 and 46.*

Savic, S., Rakovic, M., Borovac, B., and Nikolic, M. (2016). Hybrid Motion Control of Humanoid Robot for Leader-Follower Cooperative Tasks. *Thermal Science*, 20(suppl. 2):549–561. *Cited on page 39.*

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., and Westling, P. (2014). High-resolution Stereo Datasets with Subpixel-accurate Ground Truth. In *Pattern Recognition: 36th German Conference, (GCPR)*, pages 31–42. *Cited on page 197.*

Schmidt, T., Newcombe, R., and Fox, D. (2017). Self-Supervised Visual Descriptor Learning for Dense Correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427. *Cited on page 54.*

Schulz, L. (2012). The Origins of Inquiry: Inductive Inference and Exploration in Early Childhood. *Trends in Cognitive Sciences*, 16(7):382–389. *Cited on pages 1 and 192.*

Sena, A., Michael, B., and Howard, M. (2019). Improving Task-Parameterised Movement Learning Generalisation with Frame-Weighted Trajectory Generation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4281–4287. *Cited on pages 19 and 34.*

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., and Levine, S. (2018). Time-Contrastive Networks: Self-Supervised Learning from Video. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1134–1141. *Cited on pages 18, 36, and 126.*

Sharma, P., Pathak, D., and Gupta, A. (2019). Third-Person Visual Imitation Learning via Decoupled Hierarchical Controller. In *Neural Information Processing Systems (NeurIPS)*, volume 32. *Cited on pages 12, 22, 35, 36, and 126.*

Shen, W., Yang, G., Yu, A., Wong, J., Kaelbling, L. P., and Isola, P. (2023). Distilled Feature Fields Enable Few-Shot Language-Guided Manipulation. In *Conference on Robot Learning (CoRL)*, volume 229, pages 405–424. *Cited on page 16.*

Sieb, M., Xian, Z., Huang, A., Kroemer, O., and Fragkiadaki, K. (2019). Graph-Structured Visual Imitation. In *Conference on Robot Learning (CoRL)*, pages 979–989. *Cited on pages 12, 24, 30, 34, 38, 65, and 126.*

Simeonov, A., Du, Y., Tagliasacchi, A., Tenenbaum, J. B., Rodriguez, A., Agrawal, P., and Sitzmann, V. (2022a). Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation. In *IEEE Intl. Conf. on Robotics and*

Automation (ICRA), pages 6394–6400. *Cited on pages 17, 19, 21, 23, 30, 33, 54, 57, and 183.*

Simeonov, A., Du, Y., Yen-Chen, L., Rodriguez, A., Kaelbling, L. P., Lozano-Perez, T., and Agrawal, P. (2022b). SE(3)-Equivariant Relational Rearrangement with Neural Descriptor Fields. In *Conference on Robot Learning (CoRL)*, pages 835–846. *Cited on pages 23 and 30.*

Smith, L., Dhawan, N., Zhang, M., Abbeel, P., and Levine, S. (2020). AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos. In *Robotics: Science and Systems (R:SS)*. *Cited on pages 12, 18, 22, and 36.*

Song, C., Liu, G., Zhang, X., Zang, X., Xu, C., and Zhao, J. (2020). Robot Complex Motion Learning Based on Unsupervised Trajectory Segmentation and Movement Primitives. *ISA Transactions*, 97:325–335. *Cited on page 45.*

Sturm, J., Stachniss, C., and Burgard, W. (2011). A Probabilistic Framework for Learning Kinematic Models of Articulated Objects. *Journal of Artificial Intelligence Research*, 41:477–526. *Cited on page 192.*

Su, Z., Kroemer, O., Loeb, G. E., Sukhatme, G. S., and Schaal, S. (2016). Learning to Switch between Sensorimotor Primitives Using Multimodal Haptic Signals. In *International Conference on Simulation of Adaptive Behavior*, volume 9825, pages 170–182. *Cited on pages 43 and 46.*

Sundaresan, P., Belkhale, S., Sadigh, D., and Bohg, J. (2023). KITE: Keypoint-Conditioned Policies for Semantic Manipulation. In *Conference on Robot Learning (CoRL)*, volume 229, pages 1006–1021. *Cited on pages 20 and 31.*

Sundermeyer, M., Mousavian, A., Triebel, R., and Fox, D. (2021). Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 13438–13444. *Cited on page 179.*

Suomalainen, M., Calinon, S., Pignat, E., and Kyrki, V. (2019). Improving Dual-Arm Assembly by Master-Slave Compliance. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 8676–8682. *Cited on page 38.*

Sutanto, G., Fernández, I. R., Englert, P., Ramachandran, R. K., and Sukhatme, G. (2021). Learning Equality Constraints for Motion Planning on Manifolds. In *Conference on Robot Learning (CoRL)*, pages 2292–2305. *Cited on page 28.*

Tang, H., Hasegawa-Johnson, M., and Huang, T. S. (2010). Toward Robust Learning of the Gaussian Mixture State Emission Densities for Hidden Markov

Models. In *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages 5242–5245. *Cited on page 45.*

Tang, L., Jia, M., Wang, Q., Phoo, C. P., and Hariharan, B. (2023). Emergent Correspondence from Image Diffusion. In *Neural Information Processing Systems (NeurIPS)*, pages 1363–1389. *Cited on pages 15, 16, 54, 61, and 69.*

Tapia, G., Colomé, A., and Torras, C. (2024). Unsupervised Trajectory Segmentation and Gesture Recognition through Curvature Analysis and the Levenshtein Distance. In *Iberian Robotics Conference (ROBOT)*, pages 1–8. *Cited on page 43.*

Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *science*, 290(5500):2319–2323. *Cited on page 28.*

Torabi, F., Warnell, G., and Stone, P. (2019a). Generative Adversarial Imitation from Observation. In *Proceedings of the ICML Workshop on Imitation, Intent, and Interaction*. *Cited on page 19.*

Torabi, F., Warnell, G., and Stone, P. (2019b). Imitation Learning from Video by Leveraging Proprioception. In *Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 3585–3591. *Cited on page 19.*

Truong, C., Oudre, L., and Vayatis, N. (2020). Selective Review of Offline Change Point Detection Methods. *Signal Processing*, 167:107299. *Cited on pages 45 and 48.*

Tsagkas, N., Rome, J., Ramamoorthy, S., Mac Aodha, O., and Lu, C. X. (2024). Click to Grasp: Zero-Shot Precise Manipulation via Visual Diffusion Descriptors. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 11610–11617. *Cited on pages 15, 16, 19, 21, 27, and 30.*

Tsai, Y.-Y., Guo, Y., and Yang, G.-Z. (2019). Unsupervised Task Segmentation Approach for Bimanual Surgical Tasks using Spatiotemporal and Variance Properties. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1–7. *Cited on pages 44, 46, 47, 48, 49, 50, and 164.*

Ureche, A. L. P., Umezawa, K., Nakamura, Y., and Billard, A. (2015). Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations. *IEEE Trans. on Robotics*, 31(6):1458–1471. *Cited on pages 18 and 34.*

Ureche, L. P. and Billard, A. (2018). Constraints Extraction from Asymmetrical Bimanual Tasks and Their Use in Coordinated Behavior. *Robotics and Autonomous Systems*, 103:222–235. *Cited on pages 38 and 40.*

Vecerik, M., Doersch, C., Yang, Y., Davchev, T. B., Aytar, Y., Zhou, G., Hadsell, R., Agapito, L., and Scholz, J. (2024). RoboTAP: Tracking Arbitrary Points for Few-Shot Visual Imitation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5397–5403. *Cited on pages 14, 18, 24, and 30.*

Vecerik, M., Regli, J.-B., Sushkov, O., Barker, D., Pevceviciute, R., Rothörl, T., Hadsell, R., Agapito, L., and Scholz, J. (2021). S3K: Self-Supervised Semantic Keypoints for Robotic Manipulation via Multi-View Consistency. In *Conference on Robot Learning (CoRL)*, pages 449–460. *Cited on page 30.*

Verduyn, A., Vochten, M., and De Schutter, J. (2024). Enhancing Motion Trajectory Segmentation of Rigid Bodies Using a Novel Screw-Based Trajectory-Shape Representation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 7179–7185. *Cited on pages 31, 46, and 47.*

von Hartz, J. O., Welschehold, T., Valada, A., and Boedecker, J. (2024). The Art of Imitation: Learning Long-Horizon Manipulation Tasks from Few Demonstrations. *IEEE Robotics and Automation Letters*, 9(12):11369–11376. *Cited on pages 40, 43, 46, 47, and 50.*

Vosylius, V. and Johns, E. (2023). Few-Shot In-context Imitation Learning via Implicit Graph Alignment. In *Conference on Robot Learning (CoRL)*, pages 3194–3213. *Cited on pages 15 and 19.*

Wächter, M. and Asfour, T. (2015). Hierarchical Segmentation of Manipulation Actions Based on Object Relations and Motion Characteristics. In *IEEE Intl. Conf. on Advanced Robotics (ICAR)*, pages 549–556. *Cited on pages 42, 43, 46, 47, 158, 161, and 172.*

Waismeyer, A., Meltzoff, A. N., and Gopnik, A. (2015). Causal Learning from Probabilistic Events in 24-month-olds: An Action Measure. *Developmental Science*, 18(1):175–182. *Cited on pages 1, 2, and 6.*

Wang, C., Fan, L., Sun, J., Zhang, R., Fei-Fei, L., Xu, D., Zhu, Y., and Anandkumar, A. (2023a). MimicPlay: Long-Horizon Imitation Learning by Watching Human Play. In *Conference on Robot Learning (CoRL)*, pages 201–221. *Cited on pages 6 and 40.*

Wang, C., Wang, R., Mandlekar, A., Fei-Fei, L., Savarese, S., and Xu, D. (2021). Generalization Through Hand-Eye Coordination: An Action Space for Learning Spatially-Invariant Visuomotor Control. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 8913–8920. *Cited on pages 22 and 32.*

Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., and Guibas, L. J. (2019). Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2637–2646. *Cited on page 16.*

Wang, Y., Zhang, M., Li, Z., Driggs-Campbell, K. R., Wu, J., Fei-Fei, L., and Li, Y. (2023b). D3Fields: Dynamic 3D Descriptor Fields for Zero-Shot Generalizable Robotic Manipulation. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*. *Cited on pages 15, 16, 19, 23, and 31.*

Wen, B., Lian, W., Bekris, K., and Schaal, S. (2022). You Only Demonstrate Once: Category-Level Manipulation from Single Visual Demonstration. In *Robotics: Science and Systems (R:SS)*. *Cited on page 16.*

Wen, C., Lin, X., So, J., Chen, K., Dou, Q., Gao, Y., and Abbeel, P. (2024). Any-point Trajectory Modeling for Policy Learning. In *Robotics: Science and Systems (R:SS)*. *Cited on pages 21, 22, and 30.*

Xiao, Y., Wang, Q., Zhang, S., Xue, N., Peng, S., Shen, Y., and Zhou, X. (2024). SpatialTracker: Tracking Any 2D Pixels in 3D Space. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 20406–20417. *Cited on pages 14 and 86.*

Xie, F. and Chowdhury, A. (2020). Deep Imitation Learning for Bimanual Robotic Manipulation. In *Neural Information Processing Systems (NeurIPS)*, volume 33, pages 2327–2337. *Cited on pages 39 and 153.*

Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., and Sridhar, S. (2022). Neural Fields in Visual Computing and Beyond. *Computer Graphics Forum*, 41(2):641–676. *Cited on pages 14 and 57.*

Xu, A. and Zheng, W.-S. (2024). Efficient and Effective Weakly-Supervised Action Segmentation via Action-Transition-Aware Boundary Alignment. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 18253–18262. *Cited on pages 41 and 42.*

Xu, R., Chu, F.-J., Tang, C., Liu, W., and Vela, P. (2021). An Affordance Keypoint Detection Network for Robot Manipulation. *IEEE Robotics and Automation Letters*, 6(2):2870–2877. *Cited on page 30.*

Yang, J., Peng, W., Li, X., Guo, Z., Chen, L., Li, B., Ma, Z., Zhou, K., Zhang, W., Loy, C. C., and Liu, Z. (2023). Panoptic Video Scene Graph Generation. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 18675–18685. *Cited on pages 41 and 42.*

Yang, J., Zhang, J., Settle, C., Rai, A., Antonova, R., and Bohg, J. (2022). Learning Periodic Tasks from Human Demonstrations. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8658–8665. *Cited on pages 18, 30, 118, and 126.*

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., and Zhao, H. (2024). Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381. *Cited on page 15.*

Ye, J., Wang, N., and Wang, X. (2023). FeatureNeRF: Learning Generalizable NeRFs by Distilling Foundation Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8962–8973. *Cited on page 16.*

Yen-Chen, L., Florence, P., Barron, J. T., Lin, T.-Y., Rodriguez, A., and Isola, P. (2022). NeRF-Supervision: Learning Dense Object Descriptors from Neural Radiance Fields. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 6496–6503. *Cited on page 56.*

Yiu, E. and Gopnik, A. (2023). Discovering New Functions in Everyday Tools by Children, Adults and LLM's. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45. *Cited on page 1.*

Yiu, E., Kosoy, E., and Gopnik, A. (2024). Transmission Versus Truth, Imitation Versus Innovation: What Children Can Do That Large Language and Language-and-Vision Models Cannot (Yet). *Perspectives on Psychological Science*, 19(5):874–883. *Cited on pages 1 and 6.*

Yuan, J., Chew, C.-M., and Subramaniam, V. (2018). Learning Geometric Constraints of Actions from Demonstrations for Manipulation Task Planning. In *IEEE Intl. Conf. on Robotics and Biomimetics (ROBIO)*, pages 636–641. *Cited on page 19.*

Zeestraten, M. J. A. (2018). *Programming by Demonstration on Riemannian Manifolds*. PhD thesis, University of Genoa, Italy. *Cited on page 206.*

Zhang, C., Xiao, W., He, T., and Shi, G. (2024a). WoCoCo: Learning Whole-Body Humanoid Control with Sequential Contacts. In *Conference on Robot Learning (CoRL)*, volume 270, pages 455–472. *Cited on page 40.*

Zhang, H., Liu, Y., and Zhou, W. (2019). Long Time Sequential Task Learning from Unstructured Demonstrations. *IEEE Access*, 7:96240–96252. *Cited on page 45.*

Zhang, J., Herrmann, C., Hur, J., Chen, E., Jampani, V., Sun, D., and Yang, M.-H. (2024b). Telling Left from Right: Identifying Geometry-Aware Semantic Correspondence. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3076–3085. *Cited on pages 60, 62, 69, and 72.*

Zhang, J., Herrmann, C., Hur, J., Polania Cabrera, L., Jampani, V., Sun, D., and Yang, M.-H. (2023). A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence. *Advances in Neural Information Processing Systems*, 36:45533–45547. *Cited on page 16.*

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. (2023). Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems (R:SS)*. *Cited on pages 39 and 153.*

Zhou, C., Loy, C. C., and Dai, B. (2022). Extract Free Dense Labels from Clip. In *Euro. Conf. on Computer Vision (ECCV)*, pages 696–712. *Cited on page 16.*

Zhou, S., Chang, H., Jiang, S., Fan, Z., Zhu, Z., Xu, D., Chari, P., You, S., Wang, Z., and Kadambi, A. (2024). Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 21676–21685. *Cited on page 16.*

Zhou, Y., Do, M., and Asfour, T. (2016). Coordinate Change Dynamic Movement Primitives - A Leader-Follower Approach. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5481–5488. *Cited on page 38.*

Zhou, Y., Gao, J., and Asfour, T. (2019). Learning Via-Point Movement Primitives with Inter- and Extrapolation Capabilities. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4301–4308. *Cited on pages 64, 68, and 98.*

Zhou, Y., Gao, J., and Asfour, T. (2020). Movement Primitive Learning and Generalization: Using Mixture Density Networks. *IEEE Robotics & Automation Magazine*, 27(2):22–32. *Cited on pages 18, 19, and 34.*

Zhu, J., Gienger, M., and Kober, J. (2022). Learning Task-Parameterized Skills from Few Demonstrations. *IEEE Robotics and Automation Letters*, 7(2):4063–4070. *Cited on page 34.*

Zhu, Y., Jiang, Z., Stone, P., and Zhu, Y. (2023). Learning Generalizable Manipulation Policies with Object-Centric 3D Representations. In *Conference on Robot Learning (CoRL)*, volume 229, pages 3418–3433. *Cited on pages 12 and 19.*

Ziaeetabar, F., Kulvicius, T., Tamosiunaite, M., and Wörgötter, F. (2018). Recognition and Prediction of Manipulation Actions Using Enriched Semantic Event Chains. *Robotics and Autonomous Systems*, 110:173–188. *Cited on page 172.*