IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 March 2026, accepted 31 March 2026, date of publication 6 April 2026, date of current version 9 April 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3681126

RESEARCH ARTICLE

On the Effect of Domain-Adversarial Supervision Placement for Cross-Sensor 6-D Pose Estimation

TOBIAS NIEDERMAIER[1,2], SARAH WEIß[2], MAHMOUD SALEM[1], CHRISTOPHER BONENBERGER[2], MAIK KNOF[2], STEFAN ELSER[2], AND MARKUS REISCHL[1]

[1]Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
[2]Institute for Artificial Intelligence, Ravensburg-Weingarten University of Applied Sciences, 88250 Weingarten, Germany

Corresponding author: Tobias Niedermaier (tobias.niedermaier@rwu.de)

ABSTRACT Domain-adversarial training is a common approach to mitigate domain shift in deep learning, yet architectural design choices such as the placement of adversarial supervision are often treated heuristically. We present a systematic empirical assessment of gradient reversal-based domain-adversarial training applied to cross-sensor 6D pose estimation, focusing on the placement of domain classification heads within a multi-modal red, green, and blue(RGB)–point cloud fusion network. Using a simplified bidirectional RGB–point cloud fusion network, we evaluate domain head placement at multiple depths, including modality-specific encoders and intermediate fusion stages, under a strict multi-source training protocol with no access to target-domain data. Experiments are conducted on controlled synthetic datasets with multiple runs per configuration and complemented by real-world cross-sensor RGB-D evaluations. Across all settings, performance differences between domain head placements are dominated by run-to-run variability, with no statistically significant advantage for any particular placement. Real-world experiments exhibit similarly small differences and qualitatively align with synthetic results. These findings indicate that, for the studied architecture, gradient reversal-based domain-adversarial training is largely insensitive to the precise placement of the domain head, suggesting that both early and late integration constitute viable design choices. This robustness provides practical flexibility in network design, supports reproducible evaluation across placements, and motivates future work on complementary adaptation mechanisms that can be combined with adversarial supervision.

INDEX TERMS 6D pose estimation, domain adaptation, cross-sensor generalization, multi-modal sensor fusion, domain-adversarial training, robot perception.

I. INTRODUCTION

Determining an object's 3D rotation and 3D translation, commonly referred to as 6D pose estimation, from visual sensor observations such as RGB or RGB-D data, is a fundamental capability for applications ranging from robotic manipulation and industrial automation to augmented reality and autonomous driving [1], [2], [3], [4], [5]. Recent years have seen substantial progress in this field, with deep learning-based methods achieving impressive accuracy on standard benchmarks. Early convolutional neural network (CNN)-based pipelines such as PoseCNN [2] demonstrated the feasibility of learning-based 6D pose estimation on the YCB-Video dataset, while subsequent approaches including DenseFusion [3], PVN3D [6], and FFB6D [1] further improved performance by tightly integrating RGB and point cloud information. However, a common assumption underlying most of these methods is that training and test data are drawn from the same domain, typically corresponding to a fixed sensor setup and environmental conditions.

In practice, this assumption is often violated. If a trained 6D pose estimator is deployed under different sensor configurations or environmental conditions, its performance can degrade significantly due to domain shift. This issue is particularly relevant in industrial and robotic settings, where sensor hardware is frequently upgraded or replaced. Many widely

The associate editor coordinating the review of this manuscript and approving it for publication was M. Anwar Hossain.

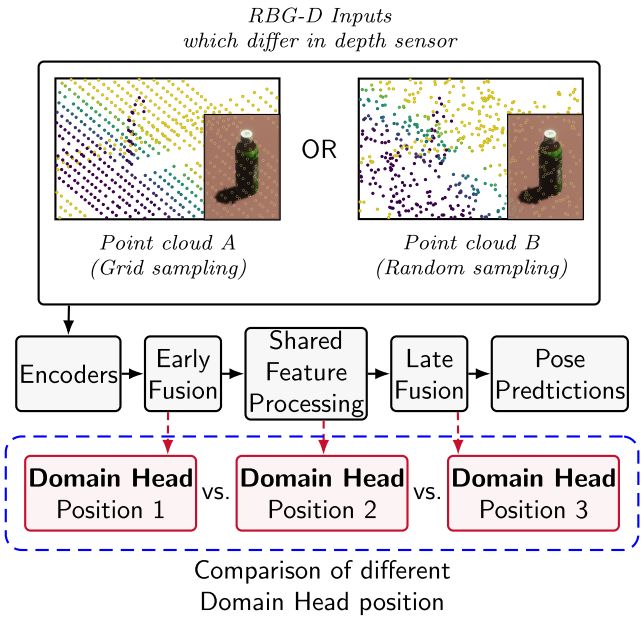

FIGURE 1. High-level overview of the architectural design choice studied in this work: a multi-modal 6D pose estimation network with domain-adversarial heads placed at different stages.

used 6D pose benchmarks, such as LineMOD and YCB-Video, were collected using older generation RGB-D sensors based on structured-light or early time-of-flight technology, such as Primesense devices (e.g. Microsoft Kinect v1/v2 or ASUS Xtion) [7], [8], [9], which are now discontinued and no longer commercially available. Alternative sensors such as Intel RealSense or Azure Kinect exhibit different noise characteristics, depth accuracy, resolution, and field-of-view properties. As a result, models trained on legacy datasets often encounter a substantial domain gap when applied to data from newer sensors, limiting their practical applicability.

Collecting and annotating new 6D pose datasets for every sensor or environment change is typically infeasible. Annotating 6-DoF object poses in real images requires precise alignment of 3D object models to sensor observations and often involves significant manual effort or complex acquisition setups [5]. Consequently, there is strong motivation to reuse existing labeled datasets and adapt models to new domains rather than repeatedly performing costly data collection and annotation. Prior work has explored synthetic data generation and domain randomization to mitigate this issue [5], [10], [11], but bridging the domain gap between training data and real deployment conditions remains a major challenge.

Domain adaptation techniques have therefore gained increasing attention in visual recognition and have been applied to 6D pose estimation [12], [13], [14]. A prominent class of methods is domain-adversarial training, as introduced by the Domain-Adversarial Neural Network (DANN) framework [12], where a domain discriminator is trained adversarially via a gradient reversal layer (GRL) to encourage the learning of domain-invariant features. Without adaptation, domain shifts caused by changes in sensors, viewpoints, or environmental conditions can lead to substantial performance degradation, with prior work reporting relative drops of 30–70% when models trained on one dataset or sensor configuration are evaluated on another [10], [15], [16], [17]. Domain-adversarial approaches have been shown to reduce dataset bias across a wide range of vision tasks, including image classification, semantic segmentation, and object detection [18], [19], [20], [21], [22], and have also been adopted in several geometry-aware and 6D pose estimation frameworks to improve cross-domain generalization without requiring additional labeled target-domain data [16], [17]. Beyond adversarial methods, alternative strategies include self-supervised domain adaptation, pseudo-labeling or self-training, and extensive domain randomization to reduce the synthetic-to-real or cross-sensor gap [10], [15], [23], [24], [25]. While these techniques differ in implementation, they share the goal of increasing robustness to domain shifts arising from changes in sensors or environments.

Despite the widespread use of domain-adversarial mechanisms, it remains unclear where domain-adaptation components should be integrated within multi-modal fusion networks. To the best of our knowledge, no prior work has systematically studied the impact of domain-head placement in 6D pose estimation networks under multi-seed and cross-domain evaluation. In practice, domain heads are attached at different stages of the network, such as image encoders, point cloud encoders, or fusion layers, often without a systematic justification. Moreover, many prior studies evaluate adaptation performance using limited experimental repetitions or without explicitly accounting for variance across random seeds, making it difficult to assess whether observed improvements are robust. Fig. 1 provides a high-level overview of the architectural design choice investigated in this work. The figure illustrates a representative multi-modal 6D pose estimation network designed to operate across different sensor configurations, exemplified here using RGB–D inputs with varying depth modalities, and highlights the alternative locations at which domain-adversarial heads can be attached.

In this work, we aim to address this gap through a systematic and statistically grounded analysis of domain-head placement in 6D pose estimation networks. Rather than introducing a new adaptation algorithm, our goal is to rigorously evaluate whether the architectural location of domain-adversarial supervision has a measurable and statistically reliable impact on cross-sensor generalization.

We focus on FFB6D-style multi-modal fusion architectures [1], representative of a class of bidirectional RGB–point cloud networks that interleave appearance and geometric features across multiple fusion stages and study the effect of inserting GRL-based domain heads at different stages of the network. To reflect realistic deployment scenarios, we adopt a cross-domain evaluation protocol inspired by the VisDA benchmark [26], in which hyperparameters are selected on a

separate validation domain and final performance is reported on a held-out test domain. This protocol helps identify minima that generalize across domains while avoiding test domain leakage and is well suited for studying adaptation behavior under domain mismatch. As such, the insights of this study are particularly relevant for researchers and practitioners developing multi-modal 6D pose estimation systems that must generalize across changing sensor configurations or deployment domains.

Our experiments span both controlled synthetic domain gaps, generated using a Unity-based simulation pipeline with systematic variations in sensor configuration, and real-world data arising from different RGB-D sensors. Across multiple random seeds and evaluation settings, we conduct a detailed statistical analysis using confidence intervals as well as independent (unpaired) hypothesis tests. Across all evaluated settings, different domain-head placements yield consistently comparable performance, with observed variations primarily attributable to run-to-run variability rather than systematic effects of placement. Furthermore, on real-world data with larger domain gaps, performance remains similarly stable across placements, with differences becoming negligible. Taken together, these results indicate that gradient reversal-based domain-adversarial training exhibits a high degree of robustness to the architectural placement of the domain head, implying that both early and late placements constitute valid design choices. This robustness underscores the importance of rigorous multi-seed evaluation and provides practical flexibility when integrating domain-adversarial components into 6D pose estimation networks.

Our main contributions are as follows:

- A systematic empirical study of domain-adversarial adaptation in multi-modal 6D pose estimation networks, focusing on the placement of GRL-based domain heads within an FFB6D-style architecture.
- A statistically grounded evaluation across multiple random seeds, using confidence intervals and hypothesis tests to assess the robustness of observed performance differences.
- A simulation-based evaluation framework with controllable sensor-induced domain gaps, enabling reproducible analysis of adaptation behavior under varying conditions.
- Experimental evidence that, under realistic cross-domain evaluation, domain-head placement does not reliably change mean performance, and that variance and stability dominate architectural micro-choices.

II. METHODOLOGY

This section describes the methodological foundation of our study. We first formalize the problem setting of cross-domain 6D pose estimation under the constraint that no target-domain data is available during training. We then introduce the base network architecture, inspired by FFB6D, and the domain-adversarial training mechanism used to encourage domain-invariant feature learning. We then detail the different domain-head placement variants investigated in this work, which constitute the central design choice evaluated in our empirical analysis. Finally, we formalize the training objective, including the pose estimation and domain-adversarial loss terms used during optimization.

A. PROBLEM SETTING AND EVALUATION UNDER UNKNOWN TARGET DOMAINS

A central challenge in deploying 6D pose estimation models in real-world robotic systems is domain shift, where the data distribution at deployment differs from that observed during training. While much of the domain adaptation literature assumes access to at least unlabeled target-domain data during training, commonly referred to as unsupervised domain adaptation, this assumption does not always hold in practice. In many industrial and robotic scenarios, the target sensor or environment is not yet available at training time, or collecting even unlabeled data from the target domain is infeasible due to hardware availability, operational constraints, or deployment timelines.

In this work, we therefore consider a stricter domain adaptation setting in which no target-domain data is available during training. The model is trained exclusively on labeled data from one or more source domains, and adaptation mechanisms must be learned without exposure to the final deployment domain. While many domain adaptation approaches in pattern recognition and visual recognition assume access to unlabeled target-domain data during training, robotic perception systems are often trained using data collected with a limited set of sensor configurations, such as standard benchmarks (e.g., YCB-Video [2]), and are subsequently deployed in environments where sensors may differ or need to be replaced. As a result, models are frequently required to generalize to previously unseen sensor setups without prior access to target-domain data.

This strict setting introduces a key methodological challenge: how to evaluate domain adaptation performance when the target domain is unknown at training time. Standard evaluation protocols commonly used in visual domain adaptation, which tune hyperparameters directly on the target domain (even without labels), are no longer applicable and risk overestimating real-world performance. Instead, model selection must rely on proxy validation signals from domains other than the target domain, encouraging generalization across domains rather than specialization to a specific deployment setting.

To address this challenge, we adopt a cross-domain validation protocol inspired by VisDA [26] for synthetic-to-real domain adaptation. In this protocol, available domains are partitioned into three disjoint sets: training domains, a validation domain, and a held-out test domain. Models are trained on the training domains, hyperparameters and architectural choices are selected based on performance on the validation domain, and final results are reported exclusively on the unseen test domain. Importantly, neither

labeled nor unlabeled data from the test domain is used during training or model selection.

This evaluation strategy serves two purposes. First, it provides a principled mechanism for model selection in the absence of target-domain data, ensuring that chosen configurations favor domain robustness rather than domain-specific overfitting. Second, it enables a fair and reproducible comparison of adaptation mechanisms by measuring their ability to generalize to genuinely unseen domains. Throughout this paper, we use this protocol to assess the effectiveness of domain-adversarial components under realistic deployment constraints.

B. BASE NETWORK ARCHITECTURE

All experiments in this work are based on a unified multi-modal fusion architecture inspired by FFB6D [1]. FFB6D combines RGB and point cloud information through multiple bidirectional fusion stages and has demonstrated strong performance on standard 6D pose estimation benchmarks. Our goal, however, is not to reproduce the full FFB6D architecture, but to adopt a simplified variant that preserves its core design principles while enabling controlled and computationally efficient experimentation.

The base network consists of four main components: an image encoder, a point cloud encoder, and a set of fusion layers that combine appearance and geometric features, followed by a pose prediction head. RGB inputs are processed by a convolutional image encoder to extract dense per-pixel features, while the point cloud input is encoded using a point-based network that produces per-point geometric descriptors. These modality-specific features are then aligned and fused to form a joint representation used for pose regression. A high-level overview of the base architecture is shown in Fig. 2, where the base network is depicted in gray and the possible domain head placements are highlighted in red (see subsubsection II-C2).

In contrast to the original FFB6D architecture, which employs multiple stacked bidirectional fusion blocks, we reduce the number of fusion layers and remove deeper recursive fusion pathways. This simplification is motivated by two considerations. First, a reduced fusion structure narrows the set of possible feature locations at which domain-adversarial components can be attached, allowing for a more controlled analysis of domain-head placement effects. Second, the full FFB6D architecture is computationally expensive, which would make the repeated multi-seed training required for statistically meaningful comparisons between architectural variants impractical. The simplified architecture therefore enables extensive multi-run evaluation while preserving the key structural components relevant to studying domain-adversarial supervision in multi-modal fusion networks.

Despite these simplifications, the overall information flow remains consistent with the FFB6D design philosophy: image and point cloud features are processed separately, aligned in a shared feature space, and fused prior to pose prediction. The pose head regresses object translation and rotation parameters from the fused features using a shared backbone across all experiments. Importantly, all architectural variants studied in this paper use the same base network; differences between variants arise solely from the presence and placement of domain-adversarial components, which are introduced in subsection II-C.

While the absolute performance of the simplified architecture differs from that of the full FFB6D model, the structural aspects relevant to this study, namely: the separation of modality-specific encoders, the presence of intermediate fusion stages, and a shared pose prediction head, are preserved. Since domain-adversarial supervision acts on intermediate feature representations and propagates gradients upstream from its attachment point, we expect the qualitative effects of domain head placement observed in this work to extend to more complex fusion-based architectures that follow the same design principles. However, architectures that employ substantially different fusion paradigms or non-modular feature hierarchies may exhibit different sensitivities to adversarial supervision placement.

By adopting a streamlined FFB6D-style architecture, we strike a balance between representational capacity and experimental tractability. This design enables systematic multi-seed evaluation and statistical analysis while retaining the essential characteristics of modern fusion-based 6D pose estimation networks.[1]

C. DOMAIN-ADVERSARIAL TRAINING AND DOMAIN HEAD PLACEMENT

This subsection introduces the domain-adversarial components used in our study and formalizes the different domain head placement strategies evaluated throughout the paper. We first describe the domain-adversarial training mechanism based on gradient reversal, which is shared across all experimental variants. We then detail the specific network locations at which the domain head is attached, which constitute the primary axis of comparison in our empirical analysis. Importantly, all variants use the same base architecture, domain discriminator, and loss formulation; they differ only in the feature level at which domain-adversarial supervision is applied.

1) DOMAIN-ADVERSARIAL TRAINING WITH GRADIENT REVERSAL

To encourage domain-invariant feature representations, we employ domain-adversarial training based on a gradient reversal layer (GRL), following the Domain-Adversarial Neural Network (DANN) framework [12]. The core idea is to introduce an auxiliary domain classification task that competes with the primary pose estimation objective. In this setup, a domain discriminator (domain head) is trained to distinguish between different domains, while the

[1] Source code is available at https://github.com/D-Doge/Domain_Head_Placement.

feature extractor is optimized to make such discrimination difficult, thereby promoting the learning of domain-invariant representations.

The training objective consists of two components. The pose estimation loss, denoted by $\mathcal{L}_{\text{pose}}$, is defined on labeled source-domain samples and supervises the prediction of object translation and rotation. This loss corresponds to the standard 6D pose regression objective used by the underlying fusion network.

In addition, a domain classification loss $\mathcal{L}_{\text{domain}}$ is applied to intermediate feature representations. The domain discriminator (domain head) predicts a discrete domain label indicating the origin domain of each sample (e.g., different simulated sensors or data sources). The domain loss $\mathcal{L}_{\text{domain}}$ is computed for all training samples and is implemented as a categorical cross-entropy objective. As a consequence, domain-adversarial training requires data from at least two distinct domains during training, such that the discriminator can learn a meaningful decision boundary between domain-specific feature distributions.

The overall training loss is given by

$$\mathcal{L} = \mathcal{L}_{\text{pose}} - \lambda(e)\,\mathcal{L}_{\text{domain}}, \tag{1}$$

where $\lambda(e) \in [0, \lambda_{\max}]$ controls the strength of the domain-adversarial signal at training epoch e. In practice, the gradient reversal layer is inserted between a feature extractor and the domain head and acts as the identity during the forward pass. During backpropagation, it multiplies the gradient of $\mathcal{L}_{\text{domain}}$ by $-\lambda(e)$ before propagating it to the feature extractor, while the domain head itself is trained to minimize $\mathcal{L}_{\text{domain}}$. As a result, the feature extractor is encouraged to produce representations that are informative for pose estimation but uninformative with respect to the domain label.

The domain-adversarial weight $\lambda(e)$ is not applied at full strength from the start, but is instead increased gradually during training. Specifically, $\lambda(e)$ follows a linear ramp-up schedule from 0 to a maximum value $\lambda_{\max}$ over the initial training epochs, after which it remains constant for the remainder of training. This schedule allows the network to first learn task-relevant pose features before progressively enforcing domain invariance, improving optimization stability and avoiding premature suppression of discriminative representations.

Across all experiments, the domain discriminator architecture, domain loss formulation, and the ramp-up schedule including the maximum weight $\lambda_{\max}$ remain unchanged. Only the network location at which the domain-adversarial loss is applied varies between configurations, as described in the following subsection.

2) DOMAIN HEAD PLACEMENT VARIANTS

While the domain-adversarial mechanism itself is fixed, the stage of the network at which it is applied can vary. The central question addressed in this work is how the placement of the domain head within a multi-modal fusion architecture influences the learning of domain-invariant representations and, consequently, cross-domain generalization.

We evaluate four domain head placement variants, corresponding to different levels of feature abstraction. The extension of the base architecture, described in subsection II-B, to include domain-adversarial supervision is illustrated in Fig. 2. In the *image encoder* variant, the domain discriminator is attached to image-based feature representations, encouraging invariance at the level of visual appearance. In the *point cloud encoder* variant, the discriminator operates on geometric features extracted from raw 3D point data. In the *fusion1* and *fusion2* variants, domain-adversarial supervision is applied at progressively later fusion stages, after image and point cloud features have been combined into a joint representation.

An important distinction between these placements lies in the scope of parameters influenced by the domain-adversarial loss during backpropagation. Gradients originating from $\mathcal{L}_{\text{domain}}$ propagate through all network parameters upstream of the attachment point. Consequently, domain heads placed at later stages in the network (e.g., *fusion2*) affect a larger portion of the model, providing greater representational capacity and processing depth over which domain-adversarial signals can act. In contrast, earlier placements restrict the adversarial signal to a smaller subset of parameters, potentially limiting the extent to which domain invariance can be achieved.

Beyond gradient scope, domain head placement also reflects a trade-off between adaptation strength and architectural stability. Later placements may allow stronger modification of the learned representation, while earlier placements leave most of the network unchanged. From a practical perspective, earlier placements may therefore be advantageous in scenarios where models are adapted incrementally across multiple domains or where large parts of the backbone are frozen during retraining.

Furthermore, the relative effectiveness of different placements may depend on the nature of the domain shift. For example, when domain differences predominantly affect geometric properties of the input, such as point density, noise characteristics, or sensor-specific sampling patterns, one might expect a domain head placed at the point cloud encoder to be more effective than one attached to the image encoder. Conversely, shifts primarily affecting visual appearance may be better addressed at earlier image-based feature levels. These considerations motivate our systematic evaluation of domain head placement under controlled conditions, without assuming that any single placement strategy is universally optimal.

In many domain-adversarial architectures [12], the domain classifier is attached to the final shared backbone representation, such that the reversed gradient influences most or all upstream feature extraction layers. In contrast, this study focuses specifically on placements within the RGB–point cloud fusion pipeline. This choice was made because the fusion stages constitute the part of the network where

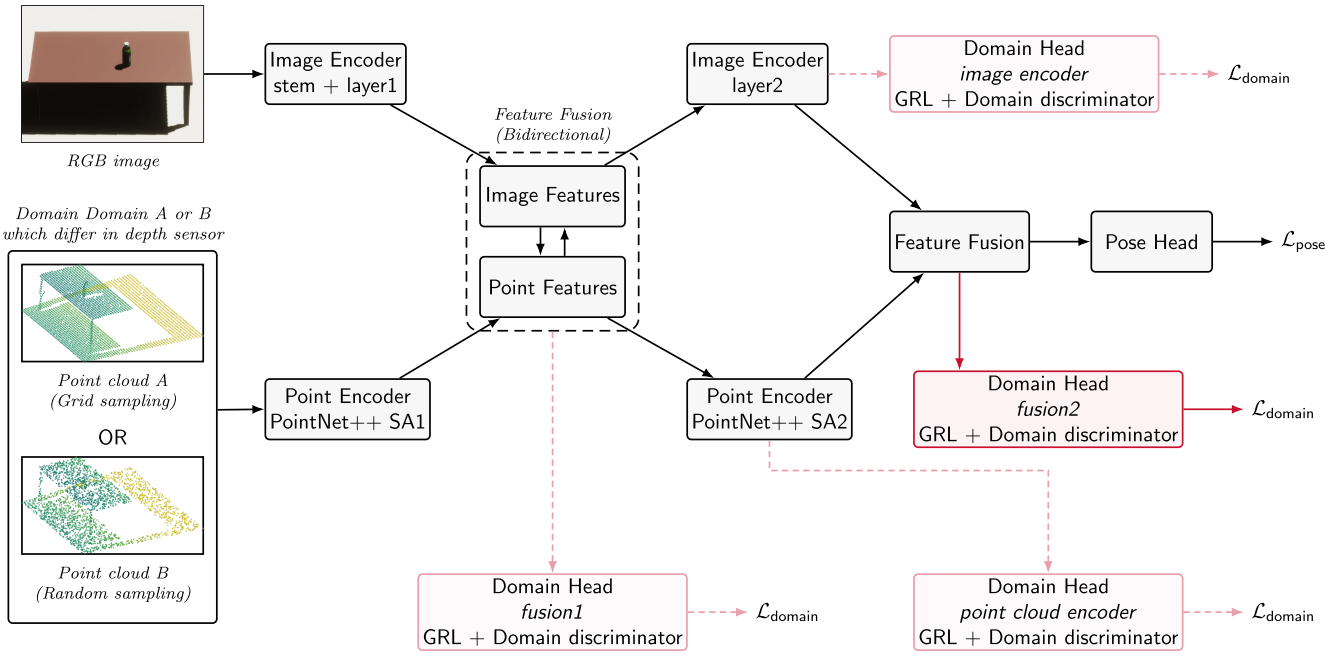

FIGURE 2. Schematic overview of the evaluated domain head placement variants. The base architecture described in subsection II-B is shown together with all possible domain-adversarial head attachment points. For illustration, the *fusion2* placement is highlighted as the active domain head, while all other candidate placements are shown using dashed arrows and a less intense color. In each experiment, exactly *one* domain head is active and contributes a single $\mathcal{L}_{\text{domain}}$ term. All other placements are disabled. During training, inputs from at least two distinct source domains are used, illustrated here as point clouds originating from different depth sensors (Point cloud A and Point cloud B), resulting in Domain A and Domain B. All variants share the same backbone architecture and domain discriminator;only the attachment point of the domain head differs.

modality-specific features are combined into a shared representation, making them a conceptually meaningful location for enforcing domain invariance. Placements directly at the task head were not considered, as features at that stage are already strongly specialized for pose prediction and are therefore less suitable for studying domain alignment within the representation-learning process. In addition, the evaluated placements were restricted to those supported by the simplified fusion architecture used in this work.

D. TRAINING OBJECTIVE

During training, the network is supervised using a multi-term pose estimation loss following an FFB6D-style dense regression formulation. The total pose loss is defined as

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{trans}} + \mathcal{L}_{\text{rot}}, \tag{2}$$

where $\mathcal{L}_{\text{obj}}$ denotes an objectness loss, $\mathcal{L}_{\text{trans}}$ a translation regression loss, and $\mathcal{L}_{\text{rot}}$ a rotation regression loss [1], [27].

No explicit weighting coefficients are applied between these loss terms. While translation errors and rotation errors exist on different numerical scales, the same loss formulation is used across all experiments to ensure a consistent comparison between domain-head placement configurations. Consequently, the study focuses on relative performance differences between architectural variants rather than on optimizing the absolute balance between translation and rotation errors.

The objectness loss $\mathcal{L}_{\text{obj}}$ supervises whether a spatial location corresponds to a valid object instance and serves to suppress background regions during dense pose regression. Following [28], we implement $\mathcal{L}_{\text{obj}}$ as a focal loss applied to dense objectness logits over the spatial prediction grid.

Let $p_i \in [0, 1]$ denote the predicted objectness probability at grid cell i and $y_i \in \{0, 1\}$ the corresponding ground-truth label, indicating object presence or background. The focal loss is defined as

$$\mathcal{L}_{\text{obj}} = -\sum_i \begin{cases} \alpha(1-p_i)^\gamma \log(p_i), & y_i = 1, \\ (1-\alpha)p_i^\gamma \log(1-p_i), & y_i = 0, \end{cases} \tag{3}$$

where α and γ are weighting and focusing parameters that reduce the impact of easy background examples and mitigate foreground or background class imbalance.

The translation loss $\mathcal{L}_{\text{trans}}$ is computed using a Smooth L1 loss L_1^*, as defined in [29] and used in [1] and [30], on the predicted 3D translation vectors. Following the dense regression formulation of FFB6D, translation is predicted at each spatial location of the network's output grid, which corresponds to discretized locations in the input observation.

Let $\hat{\mathbf{t}}_i \in \mathbb{R}^3$ and $\mathbf{t}_i \in \mathbb{R}^3$ denote the predicted and ground-truth translations at grid cell i. The loss is evaluated only at positive object locations:

$$\mathcal{L}_{\text{trans}} = \sum_{i \in \mathcal{P}} L_1^*\left(\hat{\mathbf{t}}_i - \mathbf{t}_i\right), \tag{4}$$

where $\mathcal{P}$ denotes the set of grid cells whose receptive fields overlap with ground-truth object instances and $L_1^*(\cdot)$ denotes the element-wise Smooth L1 (Huber) loss, which behaves quadratically for small residuals and linearly for large residuals.

The rotation loss $\mathcal{L}_{\text{rot}}$ is defined as a quaternion geodesic loss measuring the angular distance between predicted and ground-truth rotations on $SO(3)$. Let $\hat{\mathbf{q}}_i$ and $\mathbf{q}_i$ be unit quaternions representing the predicted and ground-truth rotations at grid cell i. The loss is given by

$$\mathcal{L}_{\text{rot}} = \sum_{i \in \mathcal{P}} 2 \arccos\left(\left|\langle \hat{\mathbf{q}}_i, \mathbf{q}_i \rangle\right|\right), \tag{5}$$

where the absolute value accounts for the antipodal symmetry of quaternion representations. This formulation is equivalent to the rotation loss used in [1].

During evaluation, the pose loss is computed per batch by aggregating the three loss components described above. Losses are averaged across all evaluated batches and, in the multi-GPU setting, across all distributed workers. The reported pose loss therefore reflects the mean per-batch loss over the entire evaluation dataset.

III. EXPERIMENTS

The goal of this experimental study is to systematically analyze the effect of domain-adversarial supervision placement within a multi-modal 6D pose estimation network. Rather than proposing a new architecture or optimizing absolute performance, we focus on isolating where gradient reversal based domain heads are applied and how this choice influences training stability, variance, and generalization under sensor domain shifts.

A. EXPERIMENTAL SETUP

All experiments are conducted using the same base network architecture and training procedure described in Section II. To isolate the effect of domain-adversarial supervision placement, the architecture, optimization parameters, data preprocessing, and training schedule are kept identical across all experimental variants. The only factor varied between runs is the location at which a single gradient reversal based domain head is attached.

Domain-adversarial training is realized by adding one domain classification head at one of four predefined locations within the network: the image encoder, the point cloud encoder, the early fusion stage (*fusion1*), or the late fusion stage (*fusion2*). In each experiment, exactly one domain head is active, producing a single domain classification loss term $\mathcal{L}_{\text{domain}}$ that competes with the pose estimation loss $\mathcal{L}_{\text{pose}}$. No configurations with multiple simultaneous domain heads are considered.

Training is performed under a multi-source domain setting, requiring samples from at least two distinct source domains. Importantly, no data from the target domain is used during training, even in an unlabeled form. This experimental setup is therefore stricter than conventional unsupervised domain adaptation and follows the evaluation philosophy of VisDA-style benchmarks [26], where models are trained on one or more known source domains and evaluated on a previously unseen target domain.

This protocol closely reflects real-world deployment scenarios in 6D pose estimation, in which models are trained on large-scale datasets collected with specific sensor configurations (e.g., YCB-Video [2]), but are later applied using different and often unknown sensor setups without the possibility of target-domain retraining or adaptation.

To account for stochasticity in network initialization and optimization, each experimental configuration in the synthetic setting is trained multiple times using different random seeds. All reported synthetic results aggregate performance across these independent runs, allowing us to assess both mean behavior and run-to-run variance.

B. MODEL SELECTION AND EARLY STOPPING

Model selection in this work refers specifically to hyperparameter-free checkpoint selection, namely the choice of the training epoch at which optimization is stopped. Ideally, one would select the checkpoint that achieves the lowest loss on the final target domain. However, since target-domain data is unavailable during training, this selection must be approximated using a separate validation domain. Checkpoint selection is performed exclusively based on validation-domain performance, without any access to or feedback from the test domain. Since continued training is expected to increasingly overfit the source training domains, a validation domain that is not used for gradient updates is employed exclusively for early stopping and checkpoint selection. Final performance is then reported on a fully unseen test domain. Early stopping is implemented with a patience of four epochs, meaning that training is terminated and the best-performing checkpoint is selected if the validation-domain loss does not improve over four consecutive epochs [31]. This strategy provides a practical stopping criterion in the absence of target-domain feedback and reflects realistic deployment scenarios in which neither labeled nor unlabeled data from the final test domain is available during training.

To account for the gradual introduction of domain-adversarial supervision, the domain-adversarial weight $\lambda(e)$ is linearly increased from zero to its maximum value $\lambda_{\max} = 0.05$ during the first five training epochs and remains constant thereafter. The value of $\lambda_{\max}$ was selected empirically based on preliminary experiments and kept fixed across all configurations to ensure a consistent comparison between domain head placements. As losses observed during this ramp-up phase are influenced by transient optimization effects and do not reflect the steady-state behavior of the domain-adversarial objective, checkpoint selection is restricted to epochs $e \geq 5$.

Restricting the search for the validation minimum to epochs at which $\lambda(e) = \lambda_{\max}$ ensures that selected

checkpoints reflect the full effect of domain-adversarial training rather than the early pose-dominated regime. Early stopping additionally provides guidance on when to terminate training in domain-adversarial setups, where the relative strength of the domain classification loss is intentionally introduced in a gradual manner.

1) EVALUATION METRIC

To evaluate the effectiveness of validation-domain-based checkpoint selection, we quantify how closely the selected checkpoint matches the optimal performance achievable on the unseen test domain. For each training run, we compute the difference

$$\Delta = \mathcal{L}_{\text{test}}(e_{\text{val}}) - \min_{e} \mathcal{L}_{\text{test}}(e), \tag{6}$$

where e_{val} denotes the epoch selected based on the minimum validation-domain loss, and $\mathcal{L}_{\text{test}}(e)$ is the pose loss evaluated on the test domain at epoch e.

This metric measures the performance gap between the checkpoint selected using validation-domain early stopping and the best-performing checkpoint on the test domain that could be identified only with hindsight. A value of $\Delta = 0$ indicates that validation-based selection successfully identifies the optimal checkpoint for the test domain, whereas larger values of Δ reflect increasingly suboptimal checkpoint selection. Importantly, Δ is always non-negative by construction.

By evaluating Δ across multiple independent runs, this metric provides insight into the reliability of validation-domain-based checkpoint selection under domain shift. While minimizing Δ does not imply guaranteed optimal test-domain performance for individual runs, consistently small values suggest that, in the considered experimental setting, the validation domain provides a useful empirical signal for guiding model selection in the absence of target-domain feedback.

C. SYNTHETIC DATASET AND EXPERIMENTAL DESIGN

For controlled analysis, we first conduct experiments on a synthetic multi-domain dataset generated using Unity [32]. The dataset comprises four distinct sensor domains, each providing paired RGB images, point cloud observations, and ground-truth 6D object pose annotations. The domains differ exclusively in the point cloud generation process, which uses different sampling schemes while keeping the RGB modality fixed. Across all domains, object geometry, scene layout, pose distributions, and annotation quality are kept identical, ensuring that performance differences arise solely from changes in the sensing configuration.

This design introduces a single, well-defined domain shift and allows us to focus specifically on the effect of point cloud sensor variation. In particular, it enables targeted analysis of whether domain-adversarial supervision placed at the image encoder or the point cloud encoder exhibits different behavior when only one modality undergoes domain change.

This controlled setup isolates the interaction between domain-adversarial supervision and the affected modality, allowing differences between domain head placements to be studied without confounding factors. In real-world settings, however, both RGB and point cloud modalities may experience domain shifts simultaneously, which typically increases the overall domain gap and may further complicate adaptation. Simultaneous shifts in both modalities would introduce multiple interacting domain factors, making it difficult to attribute observed effects to specific architectural components. Investigating such multi-modality domain shifts remains an important direction for future work.

The first domain employs structured, uniform angular sampling, acquiring points on a regular angular grid with constant inter-beam spacing, and is used as a training domain. The second domain relies on unstructured, stochastic sampling, collecting points at random spatial locations to approximate noisier or less structured sensing conditions, and is also used as a training domain. Both training domains contain an equal number of samples.

The third domain simulates the scan-line sampling pattern of a Velodyne VLP-16 sensor [33] and is used as a validation domain. The fourth domain mimics the non-uniform, floral sampling pattern characteristic of a Livox MID-70 sensor [34] and is used as the test domain. Neither the validation nor the test domain is observed during training.

To align the synthetic experiments with the scale of the real-world datasets used in this work, we generate 15,000 training frames per source domain, resulting in a balanced multi-source training set. For both the validation and test domains, 2,000 frames are generated and used exclusively for model selection and final evaluation, respectively.

An overview of the synthetic sensor domains and their respective roles is provided in Table 1.

D. REAL-WORLD DATASET AND CROSS-SENSOR EVALUATION

To validate whether trends observed on synthetic data transfer to realistic conditions, we repeat the experiments on real RGB-D datasets. Training is performed jointly on YCB-Video [2] and YCB-Ev 1.1 [35], which constitute two distinct source domains captured with different sensors and acquisition pipelines. These datasets exhibit substantially larger domain gaps than the synthetic setting, reflecting realistic variation in sensor characteristics, noise properties, and data distributions.

To ensure a balanced multi-source training setup, we enforce a 50/50 split between the two source domains. Since YCB-Ev 1.1 contains 13,851 frames in total, we subsample YCB-Video to match this number of frames. This balancing strategy prevents the domain-adversarial objective from being dominated by a single source domain and ensures that the domain classifier receives equally representative supervision from both domains.

TABLE 1. Overview of the synthetic sensor domains used in the controlled experiments. All domains share the same RGB camera configuration and differ only in the point cloud sensor model.

Role	RGB Sensor	Point Cloud Sensor	# Frames
Train	Fixed RGB camera	Grid-sampled depth sensor	15,000
Train	Fixed RGB camera	Random-sampled depth sensor	15,000
Validation	Fixed RGB camera	Scan line depth sensor	2,000
Test	Fixed RGB camera	Floral pattern depth sensor	2,000

Compared to the synthetic experiments, the domain shift between YCB-Video and YCB-Ev 1.1 is more pronounced, as it arises not only from differences in sensor hardware but also from dataset-specific capture conditions and preprocessing pipelines. While this larger domain gap may make it easier for the domain classifier to distinguish between source domains, it may also increase the risk of training instability or over-regularization when applying domain-adversarial supervision. This setting therefore provides a complementary perspective on the behavior of domain head placement under realistic and challenging conditions.

For model selection, data recorded using an Intel RealSense camera is used as a validation domain, while data captured with an Azure Kinect sensor serves as the unseen test domain. Neither validation nor test domain data is used during training. Ground-truth 6D object poses for both domains are obtained using a marker-based annotation procedure. Since these annotations are used exclusively for evaluation and not for training, they do not affect the learning process or the domain-adversarial objective.

Although both the training dataset (YCB-Ev 1.1) and the validation dataset (Custom YCB) were recorded using Intel RealSense sensors, the two datasets differ in lighting conditions, scene composition, and capture setup. As a result, the validation dataset still represents a distribution shift relative to the training data. In our experiments, the validation domain is used primarily for early stopping and model selection rather than as a proxy for a completely unseen sensor. The final evaluation is performed on the Azure Kinect test domain, ensuring that the reported performance remains independent of the validation data.

In contrast to the synthetic experiments, training on real-world data is computationally substantially more expensive. As a result, each domain head placement configuration is trained only once in the real-world setting. In the synthetic experiments, by comparison, each configuration is repeated across multiple independent runs in order to obtain statistically robust estimates of performance. Consequently, the real-world experiments should be interpreted primarily as a qualitative verification of whether the trends observed in the statistically controlled synthetic analysis also appear when training on real sensor data. Final performance is reported on the Azure Kinect test domain, enabling an assessment of cross-sensor generalization under realistic deployment conditions.

An overview of the real-world datasets, sensor domains, and their respective roles is provided in Table 2.[2]

E. EVALUATION METRICS

Our primary evaluation metric is the aggregated pose loss $\mathcal{L}_{\text{pose}}$(defined in subsection II-D). Using the pose loss as the evaluation metric ensures direct consistency between the optimization objective and reported performance, and avoids metric-dependent bias when comparing different domain head placement variants. The domain classification loss is excluded, as it serves solely as a regularization signal during training and does not correspond to task performance.

Overall, focusing on the aggregated pose loss provides a stable and objective measure for comparing domain head placement strategies across different datasets and domain shifts, while remaining directly aligned with the network's training objective.

IV. RESULTS

This section presents the empirical results of our domain-adversarial adaptation study. We first report results obtained on the synthetic multi-domain dataset, where multiple training runs per configuration enable quantitative comparison between domain head placement variants under controlled conditions. We then report results on real-world cross-sensor data, which qualitatively assess whether trends observed in the synthetic setting persist under realistic domain shifts. Reported results are interpreted in light of the observed variability across training runs.

A. SYNTHETIC DATASET RESULTS

This subsection presents the experimental results obtained on the synthetic multi-domain dataset, which enables controlled evaluation of domain-adversarial supervision placement under reproducible sensor variations. The synthetic setting allows multiple independent training runs per configuration, facilitating a statistically grounded analysis of performance differences between domain head placements. We first describe the statistical evaluation protocol used for the synthetic experiments, followed by a quantitative comparison of the resulting pose estimation performance.

1) STATISTICAL ANALYSIS PROTOCOL

To assess the robustness of observed performance differences between domain head placement variants, we employ explicit

[2]The dataset is available at https://www.kaggle.com/datasets/tobiasrwu/azure-kinect-and-realsense-rgb-d-pose-dataset

TABLE 2. Overview of real-world datasets and sensor domains used for cross-sensor evaluation.

Role	Dataset	RGB-D Sensor	# Frames
Train	YCB-Video	Asus Xtion	∼15,000
Train	YCB-Ev 1.1	Intel RealSense	∼15,000
Validation	Custom YCB	Intel RealSense	∼2,000
Test	Custom YCB	Azure Kinect	∼2,000

statistical analysis for experiments conducted on the synthetic dataset by conducting multiple independent runs per configuration and evaluating performance using confidence intervals and hypothesis testing Owing to the substantially higher computational cost of real-world training, statistical testing is limited to the synthetic setting, where multiple independent runs per configuration are available.

For synthetic experiments, each domain head placement is trained multiple times using different random seeds. For each run, the pose loss $\mathcal{L}_{\text{pose}}$ on the unseen test domain is recorded. Reported performance values correspond to the mean pose loss across runs, accompanied by measures of dispersion to quantify run-to-run variability. Since the goal of this study is to compare the relative effects of architectural design choices rather than to report absolute pose estimation accuracy, the training objective itself serves as a task-consistent and unbiased evaluation metric.

To compare domain head placement variants, we perform pairwise statistical tests on the synthetic results using Welch's t-test [36] for independent (unpaired) samples. All statistical tests are conducted at a significance level of $\alpha = 0.05$.

Confidence intervals are reported at the 95% level to convey the uncertainty associated with estimated mean differences. Importantly, absence of statistical significance is interpreted as lack of evidence for a systematic performance difference, rather than evidence of equivalence between configurations.

Overall, this statistical methodology is designed to characterize run-to-run variability and to support cautious interpretation of performance differences arising from stochastic training effects.

2) QUANTITATIVE RESULTS

To quantitatively assess whether these differences are statistically meaningful, we perform pairwise Welch's t-tests on the synthetic results (Table 3). No comparison, between two domain head positions, reaches statistical significance at the $\alpha = 0.05$ level. In particular, although the mean difference between the image encoder and point cloud encoder placements is relatively large with 0.896 the corresponding p-value is $0.076 \geq \alpha$ and the associated 95% confidence interval includes zero; consequently, the null hypothesis cannot be rejected. Overall, these results indicate that run-to-run variability dominates the effect of domain head placement in the controlled synthetic setting.

Fig. 3 summarizes the distribution of pose loss $\mathcal{L}_{\text{pose}}$ across multiple training runs for each domain head placement on the synthetic dataset. Each box represents the median and interquartile range, while individual points indicate results from independent training runs; diamonds denote the mean pose loss for each configuration. For reference, the purple dashed line indicates the performance of the base network architecture described in subsection II-B, trained exclusively on the training domains and evaluated on the target domain, representing a baseline for poor generalization. In contrast, the teal dash-dotted line corresponds to the performance of the same base network when trained directly on the test domain, serving as an oracle reference for favorable performance.

Fig. 3 summarizes the distribution of pose loss across multiple training runs for each domain head placement on the synthetic dataset. Each box represents the median and interquartile range, while individual points indicate results from independent training runs. The purple dashed line indicates the performance of the base network architecture described in subsection II-B, trained exclusively on the training domains and evaluated on the target domain, and is included solely as a reference baseline. The teal dash-dotted line corresponds to the performance of the same base network when trained directly on the test domain and serves as an oracle reference for favorable performance.

Both reference runs are provided for contextual comparison only, as repeating these training regimes multiple times for inclusion in the statistical analysis would be computationally prohibitive. Most domain-adversarial configurations outperform the baseline without a domain head. However, individual runs occasionally underperform, which may be attributable to stochastic training effects or suboptimal checkpoint selection. The majority of runs lie between the baseline and oracle references, while several runs approach oracle-level performance, with one configuration (fusion2) even exceeding the oracle reference in a single run. At the same time, fusion2 also exhibits runs that perform worse than the baseline, highlighting that while domain-adversarial training can substantially improve cross-domain adaptation, performance variance remains a significant factor.

Across all placements, we observe substantial run-to-run variability and considerable overlap between distributions. While differences in mean pose loss are visible, particularly between the image encoder and point cloud encoder placements, these differences are small relative to the observed variance. The point cloud encoder placement exhibits the lowest mean pose loss and the smallest variability, whereas the image encoder placement shows a higher mean loss. Fusion-based placements exhibit larger variance, though

TABLE 3. Pairwise Welch's t-test results between domain head placements on the synthetic dataset.

Comparison	Mean Diff.	95% CI	p-value
fusion1 vs fusion2	-0.150	[-2.273, 1.973]	0.862
fusion1 vs image encoder	-0.569	[-2.689, 1.552]	0.475
fusion1 vs point cloud encoder	0.327	[-1.938, 2.593]	0.658
fusion2 vs image encoder	-0.419	[-1.957, 1.120]	0.540
fusion2 vs point cloud encoder	0.477	[-1.009, 1.963]	0.456
image enc. vs point cloud encoder	0.896	[-0.120, 1.912]	0.076

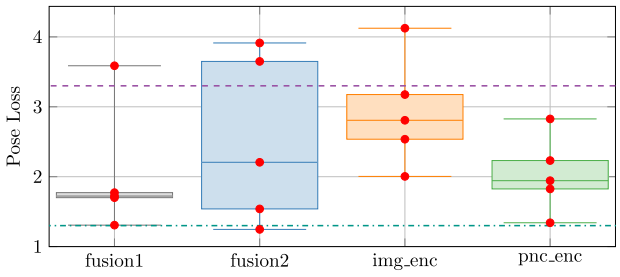

FIGURE 3. Distribution of pose loss $\mathcal{L}_{\text{pose}}$ across multiple training runs on the synthetic dataset. Boxes indicate the median and interquartile range, dots represent individual runs, and diamonds denote the mean. The purple dashed and teal dash-dotted lines correspond to baseline and oracle references, respectively (see subsubsection IV-A2).

these observations should be interpreted cautiously due to the smaller number of runs. Taken together, these results provide no statistical evidence that any particular domain head placement yields systematically better performance than the others. Rather, all evaluated placements appear to be viable design choices within the studied architecture. The broader implications of this finding are discussed in Section V.

B. REAL-WORLD CROSS-SENSOR RESULTS

For real-world experiments, each configuration is trained once due to computational constraints. Consequently, no statistical hypothesis testing is performed for real-world results. These experiments are instead used to qualitatively assess whether trends observed in the synthetic setting persist under realistic cross-sensor domain shifts.

Table 4 reports the pose loss $\mathcal{L}_{\text{pose}}$ on the validation and test domains for each domain head placement in the real-world cross-sensor setting. Across all domain-adversarial configurations, differences in pose loss on the Azure Kinect test domain are small. The relative ordering of domain head placements broadly mirrors trends observed in the synthetic experiments, with point cloud encoder and fusion-based placements yielding slightly lower test loss than the image encoder placement. However, given that each configuration is evaluated using a single training run, these differences should be interpreted qualitatively rather than as statistically meaningful.

For reference, we additionally report results obtained without domain-adversarial training. Interestingly, this baseline achieves test-domain performance comparable to that of the domain-adversarial configurations. This contrasts with the synthetic experiments, where domain-adversarial supervision consistently improved cross-domain generalization. One possible explanation is that the real-world sensor gap between training and test domains is substantially larger and more heterogeneous than the synthetic domain shifts considered earlier. Under such conditions, the domain discriminator may be able to reliably identify the domain based on low-level sensor-specific cues, making it difficult for the feature extractor to learn a truly domain-invariant representation.

In this regime, adversarial training may therefore provide limited benefit, as the optimization pressure induced by the domain loss can be satisfied by emphasizing sensor-specific characteristics rather than suppressing them. As a result, the network may effectively prioritize task performance over domain invariance, leading to similar outcomes with and without domain-adversarial supervision. We emphasize that this interpretation is speculative and not directly validated in the present study, but it is consistent with known failure modes of domain-adversarial learning under large or complex domain shifts.

Interestingly, pose loss on the test domain is consistently lower than on the validation domain across all configurations. This suggests that validation-domain-based checkpoint selection yields reasonable generalization to an unseen target sensor, even when absolute loss levels differ between domains. Overall, the real-world experiments indicate that while domain-adversarial training does not degrade performance under realistic sensor shifts. However, its effectiveness appears less pronounced when the underlying domain gap is large. Since each configuration is trained only once in the real-world setting due to computational constraints, this observation should be interpreted cautiously, as the run-to-run variability observed in the synthetic experiments may also influence the real-world results.

TABLE 4. Pose loss $\mathcal{L}_{\text{pose}}$ on validation and test domains for real-world cross-sensor experiments. Lower values indicate better performance.

Domain Head Placement	Validation	Test
fusion1	0.718	0.706
fusion2	0.731	0.695
image encoder	0.733	0.682
point cloud encoder	0.728	0.692
no domain head	0.747	0.685

C. VALIDATION DOMAIN CHECKPOINT SELECTION ANALYSIS

In addition to comparing final pose loss values, we analyze the effectiveness of the validation-domain-based

checkpoint selection and early stopping protocol described in subsection III-B in terms of its ability to yield reasonable performance on the unseen test domain. Specifically, for each training run on the synthetic dataset, we compute the performance gap Δ as defined in subsubsection III-B1.

Fig. 4 visualizes the distribution of Δ across all runs. The median gap is $\Delta = 0.35$, with an interquartile range of $[0.08, 1.238]$, indicating that validation-domain early stopping typically yields small selection gaps. While several runs exhibit near-zero gaps, a small number of runs show substantially larger deviations. This behavior is expected under domain shift, where validation and test losses are not perfectly correlated, and highlights the impact of stochastic training effects on checkpoint selection.

Fig. 5 shows an example training run, illustrating the pose loss $\mathcal{L}_{\text{pose}}$ curves on the validation and unseen test domains and the resulting checkpoint selection based on validation-domain early stopping. The pose loss curves on both domains are noisy and do not decrease monotonically, as the network is not trained on either the validation or test domain. As training progresses, the model increasingly overfits the source training domains, while domain-adversarial supervision can counteract this effect only within a limited training window. Importantly, despite differences in absolute loss scale, the validation and test domain losses exhibit a clear temporal correlation, which is essential for using the validation domain as a proxy for checkpoint selection. The consistently higher loss observed on the validation domain suggests a more challenging domain shift or lower data quality; however, this does not affect the selection procedure, as checkpoint selection relies on relative trends rather than absolute performance.

Overall, these results suggest that validation domain-based checkpoint selection provides a reasonable and practically viable strategy for model selection in the absence of target-domain data. Although it does not guarantee optimal test domain performance for every run, it avoids systematic failure and yields near-optimal checkpoints on average.

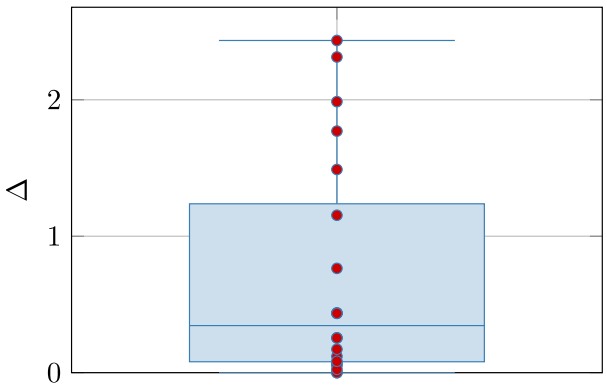

FIGURE 4. Distribution of validation–test checkpoint selection gaps Δ on the synthetic dataset. Points denote individual runs. Lower values correspond to better validation-based checkpoint selection.

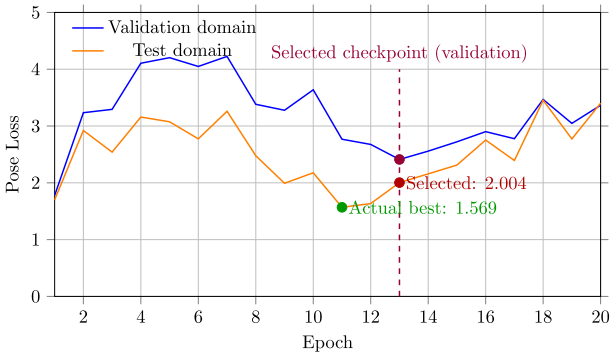

FIGURE 5. Example pose loss $\mathcal{L}_{\text{pose}}$ curves on the validation domain (blue) and unseen test domain (orange) for a run with the domain head placed at the image encoder. The checkpoint selected via validation-domain early stopping at epoch 13 (red) yields a test-domain loss of 2.004, whereas the minimum test-domain loss over all epochs is 1.569 (green), resulting in a selection gap of $\Delta = 0.435$.

D. SUMMARY OF FINDINGS

Across both synthetic and real-world experiments, we observe that the placement of gradient reversal-based domain heads has a highly variable effect on 6D pose estimation performance. In the controlled synthetic setting, where multiple independent runs per configuration enable statistical analysis, performance differences between domain head placements are dominated by run-to-run variability. No placement yields a statistically significant improvement in mean pose loss over the others, and confidence intervals show substantial overlap across all configurations.

While a weak trend toward lower mean loss and reduced variance is observed when placing the domain head at the point cloud encoder, this behavior is not statistically significant and is consistent with the experimental design, in which only the point cloud modality undergoes domain shift. Fusion-based placements exhibit comparable performance but higher variability, particularly when fewer runs are available.

Real-world cross-sensor experiments, conducted under substantially larger domain shifts and limited to single training runs per configuration, show small absolute differences in pose loss $\mathcal{L}_{\text{pose}}$ between domain head placements. The relative ordering of configurations broadly aligns with trends observed in the synthetic experiments, though these results are interpreted qualitatively due to the absence of statistical replication.

While the synthetic experiments introduce controlled domain shifts, the real-world experiments involve substantially larger sensor differences and therefore represent a more challenging adaptation scenario. Despite this increased domain gap, the qualitative trends regarding domain-head placement remain consistent with the synthetic analysis, with no clear performance advantage for any specific attachment point. In the real-world experiments, performance differences between configurations with and without domain-adversarial

supervision remain relatively small, and models trained without explicit domain alignment achieve comparable pose loss in our experiments. This observation suggests that, under the substantially larger and more heterogeneous sensor shifts encountered in real data, adversarial feature alignment alone may not always be sufficient to induce strongly sensor-invariant representations, particularly when low-level sensor characteristics remain highly discriminative.

Finally, analysis of validation domain-based checkpoint selection indicates that early stopping on a held-out validation domain provides a reasonable model selection strategy in the absence of target domain data. Although this approach does not consistently identify the test-optimal checkpoint for every run, it avoids systematic failure and yields near-optimal performance on average.

Taken together, these findings suggest that, within the studied architecture and training regime, domain head placement alone is not a dominant factor in cross-sensor generalization. Instead, stochastic variation and dataset-specific factors play a larger role, underscoring the importance of multi-run evaluation and cautious interpretation when studying domain-adversarial adaptation strategies.

The experiments in this work were conducted using a simplified bidirectional RGB–point cloud fusion architecture in order to enable extensive multi-seed evaluations. While this raises the question of whether the observed trends generalize to more complex models, the results suggest that the precise placement of the domain head within the fusion pipeline has only a limited impact on adaptation performance. One possible explanation is that adversarial supervision primarily influences the global feature distribution rather than individual architectural stages. In multi-modal fusion architectures such as the one studied here, features from different modalities are repeatedly combined across multiple layers. As a result, domain-invariant signals introduced at one stage may propagate through subsequent fusion operations, reducing the impact of the exact insertion point of the gradient reversal layer.

In addition, large domain gaps, particularly in the real-world experiments, may limit the ability of adversarial alignment to fully bridge differences in sensor characteristics. In such cases, the discriminator may primarily learn to separate domains based on low-level sensor artifacts that are difficult to suppress without also removing task-relevant information. This may help explain why adversarial supervision does not consistently improve performance under large sensor shifts. If this interpretation holds, the relative insensitivity to placement may persist in more complex architectures that employ comparable modality-fusion strategies. Nevertheless, confirming this hypothesis for deeper or more specialized architectures remains an important direction for future work.

From a practical perspective, the absence of statistically significant performance differences between domain head placements suggests that earlier attachment points in the network may be preferable. Placing the domain head closer to the input affects a smaller subset of network parameters, which can reduce the computational cost of adversarial training and limit the extent of gradient propagation through the backbone. Moreover, enforcing domain invariance at earlier feature stages may be advantageous in incremental adaptation scenarios, where large portions of the network are frozen when adapting to new sensors. Earlier placements can also improve architectural modularity, as the domain-adversarial component can be integrated as a relatively self-contained module that interacts with a limited portion of the network. This may simplify experimentation and facilitate the reuse of pretrained backbone components when adapting models across different sensor configurations. While these potential benefits are not directly evaluated in the present study, they follow naturally from the observed performance equivalence across placements and highlight a pragmatic design consideration for future domain-adversarial architectures.

V. CONCLUSION

In this work, we conducted a systematic empirical study of domain-adversarial training for cross-sensor 6D pose estimation, with a specific focus on the placement of gradient reversal-based domain heads within a multi-modal RGB–point cloud fusion network. Rather than proposing a new architecture, our goal was to rigorously evaluate whether the location at which domain-adversarial supervision is applied has a measurable and reliable impact on generalization under sensor-induced domain shifts.

Across controlled synthetic experiments with multiple independent training runs, we find that differences between domain head placements are dominated by run-to-run variability. While weak trends are observed, most notably a tendency toward lower mean loss when applying the domain head at the point cloud encoder, none of the evaluated placements yields a statistically significant improvement over the others. These findings highlight the importance of multi-seed evaluation when studying architectural design choices in domain-adversarial learning, as single-run results can be misleading.

Experiments on real-world RGB-D datasets with substantially larger domain gaps show similarly small absolute differences between placements. Although these results are limited to single training runs due to computational constraints and are therefore interpreted qualitatively, their relative ordering broadly aligns with trends observed in the synthetic setting. Importantly, real-world results further indicate that domain-adversarial training does not consistently improve performance over a baseline without adversarial supervision, suggesting that adversarial alignment alone may be insufficient under large or heterogeneous sensor shifts. Together, these results suggest that domain head placement alone is not a dominant factor in cross-sensor generalization within the studied architecture and training regime.

A further consideration concerns the architectural simplification applied to the original FFB6D network. To make the multi-seed experiments computationally tractable,

we removed some of the deeper recursive fusion pathways while preserving the overall bidirectional RGB–point cloud fusion structure. This simplification enabled the repeated training runs required to quantify run-to-run variability. However, it is possible that deeper or more complex fusion architectures could interact differently with domain-adversarial supervision, potentially amplifying differences between placement strategies. Evaluating whether domain-head placement behaves differently in larger or more complex fusion architectures therefore represents an interesting direction for future work.

In addition, we evaluated validation domain-based checkpoint selection as a practical model selection strategy under unknown target domains. Our analysis indicates that early stopping on a held-out validation domain yields near-optimal target-domain performance on average, despite occasional mismatches. This supports the use of validation domain model selection in realistic deployment scenarios where no target-domain data is available during training.

Overall, our results indicate that while domain-adversarial training remains a viable tool for promoting domain-invariant representations, its practical effectiveness is highly context-dependent and sensitive to stochastic variation, domain gap magnitude, and dataset-specific factors. In particular, differences between domain head placement variants are often dominated by run-to-run variability.

The observed run-to-run variability also has implications for how domain-adversarial methods should be evaluated. Our experiments show that performance differences between domain-head placements are often smaller than the variance introduced by stochastic training effects. As a result, reporting results from a single training run may lead to misleading conclusions about the effectiveness of architectural modifications. We therefore recommend that future studies report results over multiple random seeds and include measures of variability such as standard deviations or confidence intervals. While multi-seed experiments increase computational cost, they provide a more reliable estimate of expected model performance and reduce the risk of over-interpreting noise-driven improvements.

Beyond the specific architectural question studied in this work, the evaluation procedure used here may also provide a useful methodological template for analyzing architectural micro-choices in domain-adaptive perception systems. In particular, the use of datasets with controlled domain shifts allows individual architectural modifications to be studied in relative isolation. By varying specific sensor characteristics while keeping scenes and annotations consistent, it becomes easier to attribute observed performance differences to architectural factors rather than uncontrolled variations in the data distribution.

At the same time, the datasets used in this study represent only a limited subset of possible domain shifts. Real-world perception systems may encounter additional variations such as illumination changes, environmental dynamics, or sensor-specific noise characteristics. Consequently, a comprehensive benchmark for evaluating architectural design choices in domain adaptation would ideally include a broader range of domain shifts and sensing configurations. The evaluation framework used in this study can therefore be viewed as a starting point for such analyses rather than a complete benchmark.

Future work should therefore explore complementary strategies that address sensor-induced domain shift beyond the placement of adversarial supervision. One promising direction is the development of fusion mechanisms that more explicitly account for cross-modal and cross-sensor discrepancies, for example through learned alignment modules, sensor-aware normalization, or intermediate representations that reduce sensor-specific variability prior to fusion.

In addition, adaptation-efficient training protocols represent an important avenue for further investigation. Since earlier domain head placements interact with a smaller subset of network parameters, they may offer advantages in scenarios where models are adapted incrementally as new sensor data becomes available. Systematic studies of such incremental or continual adaptation settings could help clarify whether architectural choices influence retraining cost or stability.

More broadly, our findings suggest that domain head placement is not a dominant factor for final cross-sensor performance within the studied architecture. This observation implies that placement decisions can be guided by practical considerations, such as modularity or computational constraints, rather than expected accuracy gains alone. While these aspects are not explicitly evaluated in the present work, they point to potential directions for designing more flexible adaptation pipelines.

From a practical design perspective, the observed performance equivalence across domain head placements suggests that earlier attachment points in the network may be preferable. Placing the domain head closer to the input affects a smaller subset of parameters and limits the extent of adversarial gradient propagation, which can reduce computational overhead during training. Moreover, enforcing domain invariance at earlier feature stages may be advantageous in incremental adaptation scenarios, where large portions of the backbone are frozen when adapting to new sensors. While these potential benefits are not explicitly evaluated in the present study, they follow naturally from the empirical finding that later domain head placements do not yield systematic accuracy gains.

Finally, while this study deliberately isolates a single domain-adversarial head to enable controlled comparison, future work may consider combining adversarial supervision with complementary adaptation strategies, such as sensor-aware normalization schemes, explicit feature alignment modules, or consistency-based objectives. In practical adaptation pipelines, these mechanisms are often used together to address different aspects of the domain gap. The stochastic variability observed in our experiments suggests that careful evaluation becomes particularly important when multiple

adaptation components interact, as small architectural or training modifications may otherwise lead to differences that fall within the variance introduced by stochastic optimization. To ensure reproducibility, future studies combining domain-adversarial supervision with additional adaptation mechanisms should therefore evaluate configurations across multiple random seeds and report measures of variability such as standard deviations or confidence intervals. Investigating how such complementary strategies interact with multi-modal fusion architectures may ultimately lead to more robust cross-sensor adaptation pipelines than adversarial training in isolation.

REFERENCES

[1] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6D pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3002–3012.
[2] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," 2017, *arXiv:1711.00199*.
[3] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3338–3347.
[4] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4556–4565.
[5] S. Hinterstoißer, V. Lepetit, S. Ilić, S. M. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vis.*, 2013, pp. 548–562.
[6] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11629–11638.
[7] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.
[8] (2026). *RealSense AI*. Accessed: Jan. 17, 2016. [Online]. Available: https://www.realsenseai.com
[9] *Azure Kinect DK Documentation*. Accessed: Jan. 17, 2026. [Online]. Available: https://learn.microsoft.com/en-us/azure/Kinect-dk/
[10] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1082–10828.
[11] T. Ikeda, S. Tanishige, A. Amma, M. Sudano, H. Audren, and K. Nishiwaki, "Sim2Real instance-level style transfer for 6D pose estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sudan, Oct. 2022, pp. 3225–3232.
[12] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1180–1189.
[13] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 3490–3497.
[14] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," 2018, *arXiv:1809.10790*.
[15] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 23–30.
[16] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1530–1538.
[17] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1941–1950.
[18] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.
[19] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1989–1998.
[20] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jordan, 2017, pp. 1645–1655.
[21] G. Chang, W. Roh, S. Jang, D.-W. Lee, D. Ji, G. Oh, J. Park, J. Kim, and S. Kim, "CMDA: Cross-modal and domain adversarial adaptation for LiDAR-based 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 2, pp. 972–980.
[22] C. B. Rist, M. Enzweiler, and D. M. Gavrila, "Cross-sensor deep domain adaptation for LiDAR detection and segmentation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1535–1542.
[23] H. Akada, S. F. Bhat, I. Alhashim, and P. Wonka, "Self-supervised learning of domain invariant features for depth estimation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 997–1007.
[24] Y. Zou, Z. Yu, B. V. K. V. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 297–313.
[25] L. T. Triess, M. Dreissig, C. B. Rist, and J. Marius Zöllner, "A survey on deep domain adaptation for LiDAR perception," in *Proc. IEEE Intell. Vehicles Symp. Workshops (IV Workshops)*, Jul. 2021, pp. 350–357.
[26] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko, "VisDA: A synthetic-to-real benchmark for visual domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 2102–21025.
[27] T. Niedermaier, F. Berens, M. Reischl, and S. Elser, "A novel metric for 6D pose estimation: Addressing errors and false detections for more reliable evaluation," *at - Automatisierungstechnik*, vol. 73, no. 2, pp. 125–135, Feb. 2025.
[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
[29] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in Statistics: Methodology and Distribution*. Cham, Switzerland: Springer, 1992, pp. 492–518.
[30] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[31] L. Prechelt, "Automatic early stopping using cross validation: Quantifying the criteria," *Neural Netw.*, vol. 11, no. 4, pp. 761–767, Jun. 1998.
[32] (2020). *Unity Perception Package*. [Online]. Available: https://github.com/Unity-Technologies/com.unity.perception
[33] (2026). *Velodyne VLP-16 Puck LiDAR Sensor*. [Online]. Available: https://velodynelidar.com/products/puck
[34] (2026). *Livox MID-70 LiDAR Sensor*. [Online]. Available: https://www.livoxtech.com/mid-70
[35] P. Rojtberg and T. Pöllabauer, "YCB-ev 1.1: Event-vision dataset for 6DoF object pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2023, pp. 1–13.
[36] B. L. Welch, "The generalization of 'student's'problem when several different population varlances are involved," *Biometrika*, vol. 34, nos. 1–2, pp. 28–35, 1947.

TOBIAS NIEDERMAIER received the B.Sc. degree in applied computer science and the M.Sc. degree in computer science from Ravensburg-Weingarten University of Applied Sciences, Weingarten, Germany, in 2023 and 2025, respectively. He is currently pursuing the Dr.-Ing. degree in mechanical engineering with Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.

Since 2024, he has been a Research Associate with Ravensburg-Weingarten University of Applied Sciences. His research interests include deep learning for 3D perception, 6D object pose estimation, domain adaptation, and multi-modal sensor fusion, with a focus on cross-sensor generalization in robotic and industrial applications.

SARAH WEIß received the M.Sc. degree in computer science from Ravensburg-Weingarten University of Applied Sciences (RWU), Weingarten, Germany.

She is currently a Researcher with the Institute for Artificial Intelligence, RWU. Her research focuses on computer vision and applied machine learning, with particular emphasis on privacy-preserving perception, multimodal scene understanding, and human-centered artificial intelligence. Her current work investigates anonymization methods and their impact on downstream perception tasks, especially in industrial and healthcare environments. Her research interests include anonymization, computer vision, and multimodal data processing.

MAHMOUD SALEM received the B.Eng. degree in electronics and communication engineering from Thebes Higher Institute of Engineering, Cairo, Egypt, in 2014, and the M.Sc. degree in computer engineering and systems from Ain Shams University, Cairo, in 2020.

He is currently a Machine Learning and Robotics Scientist with the Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology, Karlsruhe, Germany. Previously, he was a Computer Vision Research Engineer with MSD and EL2Labs, Cairo. He has more than ten years of experience in research and development (R&D). Earlier, he also worked as a research and development and embedded-software engineer roles with various institutions in Egypt. His research interests include computer vision, deep learning, robot learning, collaborative robotics, and industry 4.0/5.0. During the past four years, he has contributed to international consortia, such as SMERF (Interreg EU Funding, Germany/EU), the HEAT Project (ZIM Funding, Germany/Taiwan), and Wertstromkinematik (KIT Internal Funding, Germany). He has also led industrial projects in USA, Malaysia, Egypt, Gulf region, and Sudan. He has authored or co-authored more than 15 peer-reviewed articles. His work has been recognized with the Best Research Paper Award at CARV 2023, Bologna; the Best Student Research Paper Award at KES-SDM 2019, Budapest; and the Best Paper Award at ACHI 2018, Rome. As an undergraduate, he secured podium places in four national robotics contests.

CHRISTOPHER BONENBERGER received the M.Eng. degree in electrical engineering from Ravensburg-Weingarten University of Applied Sciences, Germany, in 2016, and the Ph.D. degree in computer science from Ulm University, Germany, in 2025.

His research interests include machine learning, signal processing, and statistical learning theory, with a focus on time series analysis.

MAIK KNOF received the B.Sc. degree in applied computer science and the M.Sc. degree in computer science from Ravensburg-Weingarten University of Applied Sciences, Weingarten, Germany, in 2023 and January 2026, respectively.

He is currently an Academic Staff Member with the Institute for Artificial Intelligence, Ravensburg-Weingarten University of Applied Sciences. His research interests include robotics, 3D navigation, 3D SLAM, and environment perception, with a focus on autonomous robotic systems.

STEFAN ELSER received the Dipl.-Math. and Dr. rer. nat. degrees from the Eberhard Karls Universitaet Tuebingen, Germany, in 2008 and 2011, respectively.

Since 2018, he is currently a Professor with the Faculty of Electrical Engineering and Computer Science, Ravensburg-Weingarten University of Applied Sciences, Weingarten, Germany, where he is also a member of the Institute for Artificial Intelligence. His research interests include sensor fusion and machine learning.

MARKUS REISCHL received the Dipl.-Ing. and Ph.D. degrees in mechanical engineering from Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2001 and 2006, respectively.

Since 2020, he has been a Professor with the Faculty of Mechanical Engineering. He is currently the Head of the Research Group Machine Learning for High-Throughput and Mechatronics, Institute for Automation and Applied Computer Science, Karlsruhe Institute of Technology. His research interests include man, machine interfaces, image processing, machine learning, and data analytics.