Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

Multimodal Diffusion Transformer: Learning Versatile Behavior from Multimodal Goals

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, Rudolf Lioutikov
Intuitive Robots Lab, Karlsruhe Institute of Technology, Germany

***Abstract*—This work introduces the Multimodal Diffusion Transformer (MDT), a novel diffusion policy framework, that excels at learning versatile behavior from multimodal goal specifications with few language annotations. MDT leverages a diffusion-based multimodal transformer backbone and two self-supervised auxiliary objectives to master long-horizon manipulation tasks based on multimodal goals. The vast majority of imitation learning methods only learn from individual goal modalities, e.g. either language or goal images. However, existing large-scale imitation learning datasets are only partially labeled with language annotations, which prohibits current methods from learning language conditioned behavior from these datasets. MDT addresses this challenge by introducing a latent goal-conditioned state representation that is simultaneously trained on multimodal goal instructions. This state representation aligns image and language based goal embeddings and encodes sufficient information to predict future states. The representation is trained via two self-supervised auxiliary objectives, enhancing the performance of the presented transformer backbone. MDT shows exceptional performance on 164 tasks provided by the challenging CALVIN and LIBERO benchmarks, including a LIBERO version that contains less than 2% language annotations. Furthermore, MDT establishes a new record on the CALVIN manipulation challenge, demonstrating an absolute performance improvement of 15% over prior state-of-the-art methods that require large-scale pretraining and contain $10\times$ more learnable parameters. MDT shows its ability to solve long-horizon manipulation from sparsely annotated data in both simulated and real-world environments. Demonstrations and Code are available at https://intuitive-robots.github.io/mdt_policy/.**

I. Introduction

Future robot agents need the ability to exhibit desired behavior according to intuitive instructions, similar to how humans interpret language or visual cues to understand tasks. Current methods, however, often limit agents to process either language instructions [55, 64, 56] or visual goals [8, 47]. This restriction limits the scope of training to fully-labeled datasets, which is not scalable for creating versatile robotic agents.

Natural language commands offer the greatest flexibility to instruct robots, as it is an intuitive form of communication for humans and it has become the most popular conditioning method for robots in recent years [55, 56, 72]. However, training robots based on language instructions remains a significant challenge. Multi-Task Imitation Learning (MTIL) has emerged as a promising approach, teaching robot agents a wide range of skills via learning from diverse human demonstrations [31, 33]. Unfortunately, MTIL capitalizes on large, fully annotated datasets. Collecting real human demonstrations is notably time-consuming and labor-intensive.

One way to circumvent these challenges is Learning from Play (LfP) [32, 37], which capitalizes on large uncurated datasets. LfP allows for the fast collection of diverse demonstrations since it does not depend on scene staging, task segmentation, or resetting experiments [32]. Since these datasets are collected in such an uncurated way, they usually contain very few language annotations. However, most current MTIL methods require language annotations for their entire training set, leaving these methods with too few demonstrations to train effective policies. In contrast, future MTIL methods should be able to efficiently utilize the potential of diverse, cross-embodiment datasets like Open-X-Embodiment [7], with sparse language annotations. This work introduces a novel approach that efficiently learns from multimodal goals, and hence efficiently leverages datasets with sparse language annotations.

Recently, Diffusion Generative Models have emerged as an effective policy representation for robot learning [6, 47]. Diffusion Policies can learn expressive, versatile behavior conditioned on language-goals [65, 16]. Yet, none of the current methods adequately addresses the challenge of learning from multimodal goal specifications.

This work introduces a novel diffusion-based approach able to learn versatile behavior from different goal modalities, such as language and images, simultaneously. The approach learns efficiently even when trained on data with few language-annotated demonstrations. The performance is further improved by introducing two simple, yet highly effective self-supervised losses, Masked Generative Foresight (MGF) and Contrastive Latent Alignment (CLA). These losses encourage policies to learn latent features, that encode sufficient information to reconstruct partially-masked future frames conditioned on multimodal goals. Hence, MGF leverages the insight that policies benefit from informative latent spaces, which map goals to desired future states independent of their modality. Detailed experiments and ablations show that the additional losses enhances the performance of current state-of-the-art transformer and diffusion policies, with minimal computational overhead. The introduced Multimodal Diffusion Transformer (MDT) approach combines the strengths of multimodal transformers with MGF and latent token alignment. MDT learns versatile behavior capable of following instructions provided as language or image goals.

MDT sets new standards on CALVIN [37], a popular benchmark for language-guided learning from play data comprised of human demonstrations with few language annotations. Remarkably, MDT requires fewer than 10% of the trainable pa-

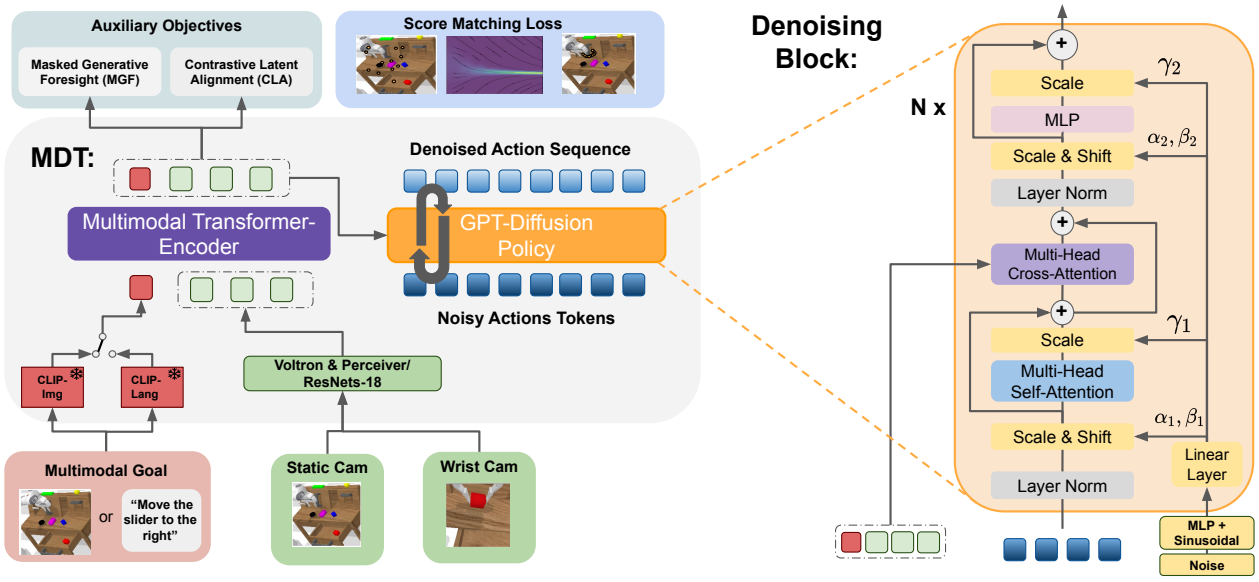


Fig. 1: (Left) Overview of the proposed multimodal Transformer-Encoder-Decoder Diffusion Policy used in MDT. (Right) Specialized Diffusion Transformer Block for the Denoising of the Action Sequence. MDT learns a goal-conditioned latent state representation from multiple image observations and multimodal goals. The camera images are processed either via frozen Voltron Encoders with a Perceiver or ResNets. The separate GPT denoising module iteratively denoises an action sequence of 10 steps with a Transformer Decoder with Causal Attention. It consists of several Denoising Blocks, as visualized on the right side. These blocks process noisy action tokens with self-attention and fuse the conditioning information from the latent state representation via cross-attention. MDT applies adaLN conditioning [42] to condition the blocks on the current noise level. In addition, it aligns the latent representation tokens of the same state with different goal specifications using self-supervised contrastive learning. The latent representation tokens are also used as a context input for the masked Image Decoder module to reconstruct masked-out patches from future images.

rameters and no additional pretraining on large-scale datasets to achieve an average 15% absolute performance gain in two CALVIN challenges. In addition, MDT performs exceptionally on the LIBERO benchmark that consists of 5 task suites featuring 130 different tasks in several environments. To show the efficiency of MDT, the tasks are modified such that only 2% of the demonstrations contain language labels. The results show that MDT is even competitive to state-of-the-art methods, that are trained on the fully annotated dataset. Through a series of experiments and ablations, the efficiency of the method and the strategic design choices are thoroughly evaluated. The major contributions of this paper are threefold:

- We introduce Multimodal Diffusion Transformer, a novel Transformer-based Diffusion approach. MDT excels in learning from multimodal goals and sets a new state-of-the-art performance on the CALVIN Challenge and across all LIBERO task suites.
- Two simple yet effective self-supervised losses for visuomotor polices to improve learning from multimodal goals. MGF and CLA improve the performance of multi-task behavior learning from sparsely labeled datasets without additional inference costs.
- A comprehensive empirical study covering over 184 different tasks across several benchmarks, verifying the performance and effectiveness of MGF and MDT.

II. Related Work

a) Language-Conditioned Robot Learning: Language serves as an intuitive and understandable interface for human-robot interactions, prompting a growing interest in language-guided learning methods within the robotics community. A growing body of work uses these models as feature generators for vision and language abstractions for downstream policy learning [56, 38, 3, 36, 33, 5, 65] and improved language expression-grounding [13, 19, 45, 66, 61]. Notably, methods like CLIPPort [54] employ frozen CLIP embeddings for language-guided pick and place, while others, such as PaLM-E [9] and RoboFlamingo [27], fine-tune vision-language models for robot control. Other methods focus on hierarchical skill learning for language-guided manipulation in LfP [37, 32, 38, 48, 71, 63]. Further, transformer-based methods without hierarchical structures [72, 8, 47, 64, 51, 49], focus on next-action prediction based on previous observation tokens. Multi-Task Action Chunking Transformer (MT-ACT), for instance, utilizes a Variational Autoencoder (VAE) transformer encoder-decoder policy, encoding only the current state and a language goal to generate future actions [3, 11, 70].

Furthermore, diffusion-based methods have gained adoption as policy representations that iteratively diffuse actions from Gaussian noise [18, 59]. Several diffusion policy approaches focus on generating plans on different abstraction levels for

behavior generation. LAD [69] trains a diffusion model to diffuse a latent plan sequence in pre-trained latent spaces of HULC [36] combined with HULC's low-level policy. UniPy [10] and AVDC [24] directly plan in the image space using video diffusion models and execute the plan with another model. Frameworks related to MDT are Distill-Down [16] and Play-Fusion [5], which also utilize diffusion policies for language-guided policy learning. Both methods use variants of the CNN-based diffusion policy [6]. However, all these methods require fully annotated datasets to learn language-conditioned policies. MDT effectively learns from multimodal goals, enabling it to leverage partially annotated datasets.

b) Self-Supervised Learning in Robotics: An increasing body of work in robotics studies self-supervised representations for robot control. A key area is learning universal vision representations or world-models, typically trained on large, diverse offline datasets using either masking strategies [22, 17, 12, 67, 50, 35] or contrastive objectives [14, 41, 52, 68, 40, 25, 2, 34, 46]. Another body of work explores robust representations for robot policies from multiple sensors, using token masking strategies [44] or generative video generation [64]. However, these methods require specific transformer models that rely on a long history of multiple states, which is a limitation for token masking and video generation techniques. Notably, Crossway-Diffusion [26] proposes a self-supervised loss specifically designed for CNN-based diffusion policies [6] by redesigning the latent space of the U-net diffusion model to reconstruct the current image observation and proprioceptive features for better single task performance.

In order to predict a sequence of future actions efficiently, some recent approaches deploy transformer-based policies that encode only the current state information without any history of prior states [3, 11, 70]. Traditional token masking strategies [44] or video generation objectives [64] that rely on token sequences of multiple states for pretraining are incompatible with such single state models, since they rely on a history of prior states. To bridge this gap, the proposed MGF and CLA objectives enhance the capabilities of these single-state observation policies. MGF and CLA enable learning of versatile behavior from multimodal goals efficiently and without additional inference costs and can also be used for pretraining on action-free data.

c) Behavior Generation from Multimodal Goals: While recent advancements in goal-conditioned robot learning have predominantly focused on language-guided methods, there is a growing interest in developing agents capable of interpreting instructions across different modalities, such as goal images, sketches, and multimodal combinations. Mutex [53] presents an imitation learning policy that understands goals in natural speech, text, videos, and goal images. Mutex further uses cross-modality pretraining to enhance the model's understanding of the different goal modalities. Steve-1 [28] is a Minecraft agent that uses a VAE encoder to translate language descriptions into the latent space of video demonstrations of the task, enabling it to follow instructions from both videos or text descriptions. Other research efforts are exploring novel conditioning methods. Examples include using the cosine distance between the current state and a goal description from fine-tuned CLIP models [50] or employing multimodal prompts [21] that combine text with image descriptions. Rt-Sktech and Rt-Trajectory present two new conditioning methods leveraging goal sketches of the desired scene [72] and sketched trajectories of the desired motion [15], respectively. While MDT primarily addresses the two most prevalent goal modalities, namely text and images, our framework is in theory versatile enough to incorporate other modalities like sketches.

III. Method

MDT is a diffusion-based transformer encoder-decoder architecture that simultaneously leverages two self-supervised auxiliary objectives. Namely Contrastive Latent Alignment and Masked Generative Foresight. First, the problem definition is provided. Next, the continuous-time diffusion formulation, essential for understanding action sequence learning from play, of MDT is discussed. Followed by an overview of the proposed transformer architecture of MDT. Afterward, the two novel self-supervised losses are introduced.

A. Problem Formulation

The goal-conditioned policy $\pi_\theta(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i, \boldsymbol{g})$ predicts a sequence of actions $\bar{\boldsymbol{a}}_i = (\boldsymbol{a}_i, \dots, \boldsymbol{a}_{i+k-1})$ of length k, conditioned on both the current state embedding $\boldsymbol{s}_i$ and a latent goal $\boldsymbol{g}$. The latent goal $\boldsymbol{g} \in \{\mathbf{o}, \mathbf{l}\}$ encapsulates either a goal-image $\mathbf{o}$ or an encoded free-form language instruction $\mathbf{l}$. MDT learns such policies from a set of task-agnostic play trajectories $\mathcal{T}$. Each individual trajectory $\tau \in \mathcal{T}$ represents a series of tuples $\tau = ((\boldsymbol{s}_1, \boldsymbol{a}_1), \dots, (\boldsymbol{s}_{T_n}, \boldsymbol{a}_{T_n}))$, with observation $\boldsymbol{s}_i$, action $\boldsymbol{a}_i$. The final play dataset is defined as $\mathcal{D} = \{(\boldsymbol{s}_i, \bar{\boldsymbol{a}}_i)|\bar{\boldsymbol{a}}_i = (\boldsymbol{a}_i, \dots, \boldsymbol{a}_{i+k-1}), (\boldsymbol{s}_i, \boldsymbol{a}_i) \in \tau, \tau \in \mathcal{T}\}$. During training, a set of goals is created for each datapoint $\mathcal{G}_{\boldsymbol{s}_i, \bar{\boldsymbol{a}}_i} = \{\mathbf{o}_i, \mathbf{l}_i\}$, where $\mathbf{l}_i$ is the language annotation for the state $\boldsymbol{s}_i$ if it exists in the dataset. The goal image $\mathbf{o}_i = \boldsymbol{s}_{i+j}$ is a future state where the offset j is sampled from the geometric distribution with bounds $j \in [20, 50]$ and probability of 0.1. MDT maximizes the log-likelihood across the play dataset,

$$\mathcal{L}_{\text{play}} = \mathbb{E}\left[\sum_{(\boldsymbol{s}_i, \bar{\boldsymbol{a}}_i) \in \mathcal{D}} \sum_{\boldsymbol{g} \in \mathcal{G}_{\boldsymbol{s}_i, \bar{\boldsymbol{a}}_i}} \log \pi_\theta\left(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i, \boldsymbol{g}\right)\right]. \quad (1)$$

Human behavior is diverse and there commonly exist multiple trajectories converging towards an identical goal. The policy must be able to encode such versatile behavior [4] to learn effectively from play.

B. Score-based Diffusion Policy

In this section, the language-guided Diffusion Policy for learning long-horizon manipulation from play with limited language annotations is introduced. Diffusion models are generative models that learn to generate new data from random Gaussian noise through an iterative denoising process. The models are trained to subtract artificially added noise with various noise levels. Both the procedures of adding and subtracting noise can be described as continuous time processes

stochastic-differential equations (SDEs) [59]. MDT leverages a continuous-time SDE formulation [23]

$$\mathrm{d}\bar{\boldsymbol{a}}_i = \big(\beta_t\sigma_t - \dot{\sigma}_t\big)\sigma_t\nabla_a \log p_t(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i,\boldsymbol{g})dt + \sqrt{2\beta_t}\sigma_t d\omega_t, \quad (2)$$

that is commonly used in image generation [23, 60]. The score-function $\nabla_{\bar{\boldsymbol{a}}_i} \log p_t(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i,\boldsymbol{g})$ is parameterized by the continuous diffusion variable $t \in [0, T]$, with constant horizon $T > 0$. This formulation reduces the stochasticity to the Wiener process ω_t, which can be interpreted as infinitesimal Gaussian noise that is added to the action sample. The noise scheduler σ_t defines the rate of added Gaussian noise depending on the current time t of the diffusion process. Following best practices [23, 47, 60], MDT uses $\sigma_t = t$ for the policy. The range of noise perturbations is set to $\sigma_t \in [0.001, 80]$ and the action range is rescaled to $[-1, 1]$. The function β_t describes the replacement of existing noise through injected new noise [23]. This SDE is notable for having an associated ordinary differential equation, the Probability Flow ODE [59]. When action chunks of this ODE are sampled at time t of the diffusion process, they align with the distribution $p_t(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i,\boldsymbol{g})$,

$$\mathrm{d}\bar{\boldsymbol{a}}_i = -t\nabla_{\bar{\boldsymbol{a}}_i} \log p_t(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i,\boldsymbol{g})\,\mathrm{d}t. \quad (3)$$

The diffusion model learns to approximate the score function $\nabla_{\bar{\boldsymbol{a}}_i} \log p_t(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i,\boldsymbol{g})$ via Score matching (SM) [62]

$$\mathcal{L}_{\text{SM}} = \mathbb{E}_{\sigma,\bar{\boldsymbol{a}}_i,\boldsymbol{\epsilon}}\big[\alpha(\sigma_t)\|D_\theta(\bar{\boldsymbol{a}}_i + \boldsymbol{\epsilon}, \boldsymbol{s}_i, \boldsymbol{g}, \sigma_t) - \bar{\boldsymbol{a}}_i\|_2^2\big], \quad (4)$$

where $D_\theta(\bar{\boldsymbol{a}}_i + \boldsymbol{\epsilon}, \boldsymbol{s}_i, \boldsymbol{g}, \sigma_t)$ is the trainable neural network. During training, noise levels from a noise distribution p_{noise} are sampled randomly and added to the action sequence and the model predicts the denoised action sequence. To generate actions during a rollout, the learned score model is inserted into the reverse SDE and the model iteratively denoises the next sequence of actions. By setting $\beta_t = 0$, the model recovers the deterministic inverse process that allows for fast sampling in a few denoising steps without injecting additional noise into the inverse process [59]. Detailed training and inference description can be found in subsection A of the Appendix. For the experiments, MDT uses the DDIM sampler [59] to diffuse an action sequence in 10 steps.

C. Model Architecture

MDT uses a multimodal transformer encoder-decoder architecture to approximate the conditional score function of the action sequence. The encoder first processes the tokens from the current image observations and desired multimodal goals, converting these inputs into a series of latent representation tokens. The decoder functions as a diffuser that denoises a sequence of future actions. Figure 1 illustrates the architecture.

First, MDT encodes image observations of the current state from multiple views with image encodings. This work introduces two encoder versions of MDT: *MDT-V*, a variant with the frozen Voltron embeddings and *MDT*, the default model with ResNets. The MDT-V encoder leverages a Perceiver-Resampler to improve computational efficiency [1]. Each image is embedded into 196 latent tokens by Voltron. The Perceiver module uses multiple transformer blocks with cross attention to compress these Voltron tokens into a total of 3 latent tokens. This procedure results in a highly efficient feature extractor that capitalizes on pretrained Voltron embeddings. The MDT encoder uses a trainable ResNet-18 with spatial softmax pooling and group norm [6] for each camera view. Each ResNet returns a single observation token for every image. Both MDT encoder versions embed goal images and language annotations via frozen CLIP models [43] per goal-modality into a single token. After the computation of the embeddings, both MDT encoders apply the same architecture comprised of several self-attention transformer layers, resulting in a set of informative latent representation tokens.

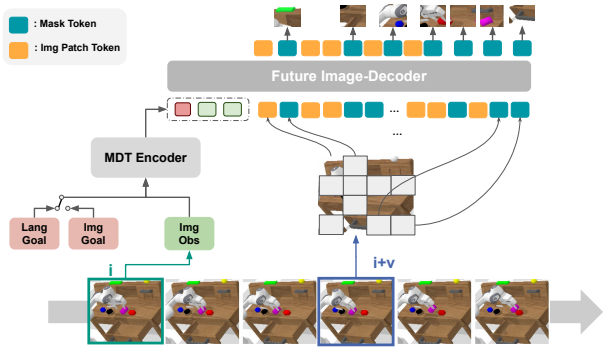


Fig. 2: The Masked Generative Foresight Auxiliary Task enhances the MDT model. It starts by encoding the current observation and goal using the MDT Encoder. The resulting latent state representations then serve as conditional inputs for the Future Image-Decoder. This decoder receives encoded patches of future camera images along with mask tokens. Its task is to reconstruct the occluded patches in future frames.

The MDT diffusion decoder denoises the action sequence with causal masking. Cross-attention in every decoder layer fuses the conditioning information from the encoder into the denoising process. The current noise level σ_t is embedded using a Sinusoidal Embedding with an additional MLP into a latent noise token. MDT conditions the denoising process to the current noise level via AdaLN-conditioning on the Transformer Decoder blocks [42]. The right part of Figure 1 illustrates this process, including all internal update steps. The proposed framework separates representation learning from denoising, resulting in a more computationally efficient model since the model only needs to encode the latent representation tokens once. Further, the experiments demonstrate that the proposed denoising model achieves higher performance than prior Diffusion-Transformer architectures [6]. MDT uses the same set of hyperparameters across all experiments.

D. Masked Generative Foresight

A key insight of this work is that policies require an informative latent space to understand how desired goals will change the robot's behavior in the near future. Consequently, policies that are able to follow multimodal goals have to map different goal modalities to the same desired behaviors.

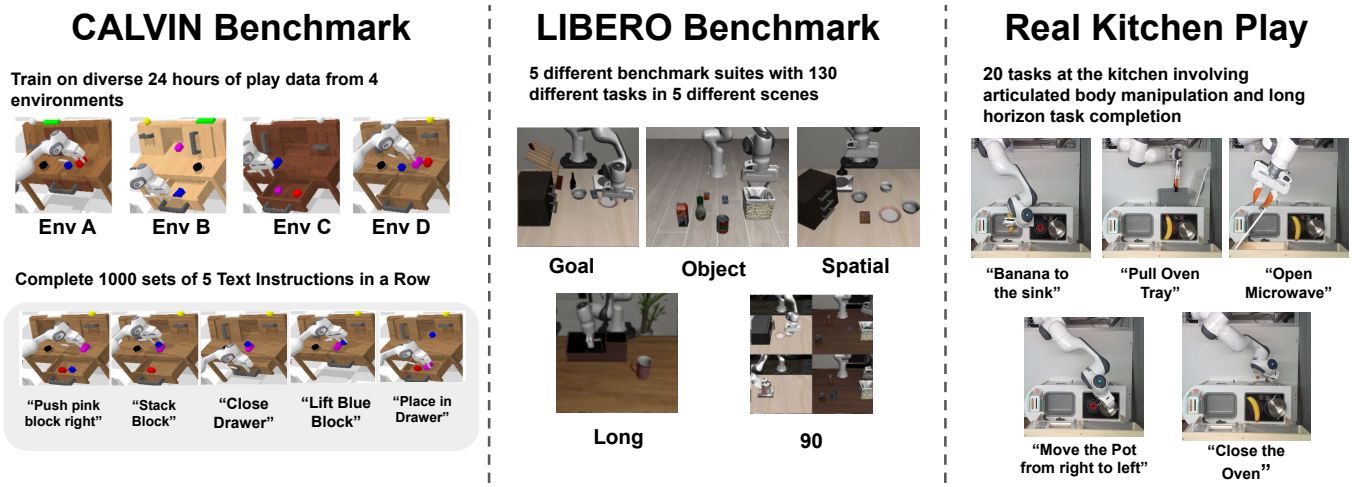


Fig. 3: Overview of the different environments used to test MDT: (Left) CALVIN Benchmark consisting of four environments each with unique positions and textures for slider, drawer, LED, and lightbulb. (Middle) Overview of the different tasks and scene diversity in the LIBERO benchmark, which is divided into 5 different task suites. (Right) Example tasks from the real robot experiments at a toy kitchen, where models are tested after training on partially labeled play data.

Whether a goal is defined through language or represented as an image, the intermediate changes in the environment are identical across these goal modalities. The proposed *Masked Generative Foresight*, an additional self-supervised auxiliary objective, builds upon this insight. Given the latent embedding of the MDT(-V) encoder for state $\boldsymbol{s}_i$ and goal $\boldsymbol{g}$, MGF trains a Vision Transformer (ViT) to reconstruct a sequence of 2D image patches $(\mathbf{u}_1, \ldots, \mathbf{u}_U) = \texttt{patch}(\boldsymbol{s}_{i+v})$ of the future state $\boldsymbol{s}_{i+v}$, with $v = 3$ being the foresight distance used across all experiments in this work. A random subset of U of these patches is replaced by a mask token. Even though the ViT now receives both masked and non-masked patches only the reconstruction of the masked patches contributes to the loss

$$\mathcal{L}_{\text{MGF}}\left(\boldsymbol{s}_i\right) = \frac{1}{U} \sum_{\mathbf{u} \in \texttt{patch}(\boldsymbol{s}_{i+v})} \mathbf{1}_{\text{m}}(\mathbf{u}) \left(\mathbf{u} - \hat{\mathbf{u}}\right)^2, \tag{5}$$

where the indicator function $\mathbf{1}_{\text{mk}}(\mathbf{u})$ is 1 if $\mathbf{u}$ is masked and 0 otherwise. Detailed hyperparameters for the model are summarized in Table V of the Appendix.

MGF differs from existing approaches [26, 64], which require full reconstruction of images or videos. While various other masking methods exist [22], all of them aim to learn robust representations of the current state, while MGF reconstructs future states to include foresight into the latent embedding. MGF is conceptually simple and can be universally applied to all transformer policies. Section IV-D shows that the advantages of MGF are not specific to MDT but also increases the performance of MT-ACT [3].

E. Aligning Latent Goal-Conditioned Representations

To effectively learn policies from multimodal goal specifications, MDT must align visual goals with their language counterparts. A common approach to retrieve aligned embeddings between image and language inputs is the pre-trained CLIP model, which has been trained on paired image and text samples from a substantial internet dataset [43]. However, CLIP exhibits a tendency towards static images and struggles to interpret spatial relationships and dynamics [66, 39, 36]. These limitations, lead to an insufficient alignment in MTIL since goal specifications in robotics are inherently linked to the dynamics between the current state $\boldsymbol{s}_i$ and the desired goal $\boldsymbol{g}$. Instead of naively fine-tuning the large 300-million-parameter CLIP model, MDT introduces an auxiliary objective that aligns the MDT(-V) state embeddings conidtioned on different goal modalities. These embeddings include the goal as well as the current state information, allowing the CLA objective to consider the task dynamics.

Since CLA requires a single vector for each goal modality, the various MDT-V latent tokens are reduced via Multi-head Attention Pooling [22] and subsequently normalized. MDT uses the embedding of the static image as a representative token to compute the contrastive loss. Hence, every training sample $(\boldsymbol{s}_i, \bar{\boldsymbol{a}}_i)$ that is paired with a multimodal goals specification $\mathcal{G}_{\boldsymbol{s}_i, \bar{\boldsymbol{a}}_i} = \{\mathbf{o}_i, \mathbf{l}_i\}$ is reduced to the vectors $\boldsymbol{z}_i^{\mathbf{o}}$ and $\boldsymbol{z}_i^{\mathbf{l}}$ for images and language goals respectively. CLA computes the InfoNCE loss using the cosine similarity $C\left(\boldsymbol{z}_i^{\mathbf{o}}, \boldsymbol{z}_i^{\mathbf{l}}\right)$ between the image-goal conditioned state embedding $\boldsymbol{z}_i^{\mathbf{o}}$ and the language-goal conditioned state embedding $\boldsymbol{z}_i^{\mathbf{l}}$

$$\mathcal{L}_{\text{CLA}} = -\frac{1}{2B} \sum_{i=1}^{B} \left(\log \left(\frac{\exp\left(\frac{C\left(\boldsymbol{z}_i^{\mathbf{o}}, \boldsymbol{z}_i^{\mathbf{l}}\right)}{v}\right)}{\sum_{j=1}^{B} \exp\left(\frac{C\left(\boldsymbol{z}_i^{\mathbf{o}}, \boldsymbol{z}_j^{\mathbf{l}}\right)}{v}\right)} \right) + \log \left(\frac{\exp\left(\frac{C\left(\boldsymbol{z}_i^{\mathbf{o}}, \boldsymbol{z}_i^{\mathbf{l}}\right)}{v}\right)}{\sum_{j=1}^{B} \exp\left(\frac{C\left(\boldsymbol{z}_j^{\mathbf{o}}, \boldsymbol{z}_i^{\mathbf{l}}\right)}{v}\right)} \right) \right), \tag{6}$$

with temperature parameter v and batch size B. The full MDT

Train	Method	No. Instructions in a Row (1000 chains)					
		1	2	3	4	5	**Avg. Len.**
D	HULC	82.5%	66.8%	52.0%	39.3%	27.5%	2.68±(0.11)
	LAD	88.7%	69.9%	54.5%	42.7%	32.2%	2.88±(0.19)
	Distill-D	86.7%	71.5%	57.0%	45.9%	35.6%	2.97±(0.04)
	MT-ACT	88.4%	72.2%	57.2%	44.9%	35.3%	2.98±(0.05)
	MDT (ours)	**93.3%**	**82.4%**	**71.5%**	**60.9%**	**51.1%**	**3.59±(0.07)**
	MDT-V (ours)	**93.7%**	**84.5%**	**74.1%**	**64.4%**	**55.6%**	**3.72±(0.05)**
ABCD	HULC	88.9%	73.3%	58.7%	47.5%	38.3%	3.06±(0.07)
	Distill-D	86.3%	72.7%	60.1%	51.2%	41.7%	3.16±(0.06)
	MT-ACT	87.1%	69.8%	53.4%	40.0%	29.3%	2.80±(0.03)
	RoboFlamingo	96.4%	89.6%	82.4%	74.0%	66.0%	4.09±(0.00)
	MDT (ours)	**97.8%**	**93.8%**	**88.8%**	**83.1%**	**77.0%**	**4.41±(0.03)**
	MDT-V (ours)	**98.6%**	**95.8%**	**91.6%**	**86.2%**	**80.1%**	**4.52±(0.02)**

TABLE I: Performance comparison of various policies learned end-to-end on the CALVIN ABCD→D and D→D challenge within the CALVIN benchmark. The table shows the average rollout length to solve 5 instructions in a row (Avg. Len.) of 1000 chains. The proposed methods MDT and MDT-V significantly outperform all reported baselines averaged over 3 seeds on both datasets and sets a sota performance.

loss combines the Score Matching loss, from Eq. (4), the MGF loss from Eq. (5) and the CLA loss from Eq. (6)

$$\mathcal{L}_{\text{MDT}} = \mathcal{L}_{\text{SM}} + \alpha\mathcal{L}_{\text{MGF}} + \beta\mathcal{L}_{\text{CLIP}}, \tag{7}$$

where $\alpha = 0.1$ and $\beta = 0.1$ in most experiment settings.

IV. Evaluation

In this section, we examine the performance of MDT on CALVIN [37] and LIBERO [29], two established benchmarks for Language-conditioned Imitation Learning. MDT is tested against several state-of-the-art methods on both benchmarks. In addition, we evaluate MDT in a real world play setting. The experiments aim to answer the following questions:

- (I) Is MDT able to learn long-horizon manipulation from play data with few language annotations?
- (IIa) Do MGF and CLA enhance the performance of MDT?
- (IIb) Does MGF improve the performance of other transformer policies?
- (III) Can MDT learn language-guided manipulation from partially labeled data in a real-world setting?

A. Evaluation on CALVIN

The CALVIN challenge [37] consists of four similar but different environments A, B, C, D. The four setups vary in desk shades and the layout of items as visualized in Figure 3. The main experiments for this benchmark are conducted on the full dataset ABCD→D, where the policies are trained on ABCD and evaluated on D. This setting contains 24 hours of uncurated teleoperated play data with multiple sensor modalities and 34 different tasks for the model to learn. Further, only 1% of data is annotated with language descriptions. All methods are evaluated on the long-horizon benchmark, which consists of 1000 unique sequences of instruction chains, described in natural language. Every sequence requires the robot to continuously solve 5 tasks in a row. During the rollouts, the agent gets a reward of 1 for completing the instruction with a maximum of 5 for every rollout. We additionally perform experiments on the small benchmark D→D consisting of only 6 hours of play data to study the data efficiency of our proposed method.

Baselines. We compare our proposed policy against the following state-of-the-art language-conditioned multi-task policies on CALVIN:

- **HULC:** A hierarchical play policy, that uses discrete VAE skill space with an improved low-level action policy and a transformer plan encoder to learn latent skills [36].
- **LAD:** A hierarchical diffusion policy, that extends the HULC policy by substituting the high-level planner with a U-Net Diffusion model [69] to diffuse plans.
- **Distill-D:** A language-guided Diffusion policy from [16], that extends the initial U-Net diffusion policy [6] with additional Clip Encoder for language-goals. We use our continuous time diffusion variant instead of the discrete one for direct comparison and extend it with the same CLIP vision encoder to guarantee a fair comparison.
- **MT-ACT:** A multitask transformer policy [3, 70], that uses a VAE encoder for action sequences and also predicts action chunks instead of single actions with a transformer encoder-decoder architecture.
- **RoboFlamingo:** A finetuned Vision-Language Foundation model [27] containing 3 billion parameters, that has an additional recurrent policy head for action prediction. The model was pretrained on a large internet-scale set of image and text data and then finetuned for CALVIN.

We adopt the recommended hyperparameters for all baselines to guarantee a fair comparison and give an overview of our chosen hyperparameters for self-implemented baselines in Appendix C. Further, we directly compare the self-reported results of HULC, LAD, and RoboFlamingo on CALVIN [69, 36, 27, 69]. All models use the same language and image goal models to ensure fair comparisons. Since RoboFlamingo only published the best seed of each model, we can not include standard deviations in their results.

Results. The results of all our experiments on CALVIN are summarized in Table I. We assess the performance of MDT and MDT-V on ABCD→D and on the small subset D→D. The results are shown in Table I. MDT-V sets a new record in the CALVIN challenge, extending the average rollout length to 4.52 which is a 10% absolute improvement over RoboFlamingo. MDT also surpasses all other tested methods. Notably, MDT achieves this while having less than 10% of trainable parameters and not requiring pretraining on large-scale datasets. In the scaled-down CALVIN D→D benchmark, MDT-V establishes a new standard, outperforming recent methods like LAD [69] and boosting the average rollout length by 20% over the second best baseline. While RoboFlamingo demonstrates commendable performance on the complete ABCD dataset, it relies on substantial training data and remains untested on the D→D subset. In contrast, MDT excels in both scenarios with remarkable efficiency.

Method	Lang. Annotation	$\mathcal{L}_{\text{CLA}}$	$\mathcal{L}_{\text{MGF}}$	Spatial	Object	Goal	Long	90	Average
Transformer-BC [29]	100 %	×	×	71.8 ± 3.7	71.00 ± 7.9	**76.3 ± 1.3**	24.2 ± 2.6	-	-
Distill-D [16]	2%	×	×	46.8 ± 2.8	72.0 ± 6.5	63.8 ± 2.5	47.3 ± 4.1	49.9 ± 1.0	56.0 ± 3.4
MDT	2%	×	×	66.0 ± 1.9	85.2 ± 2.3	67.8 ± 4.6	65.0 ± 2.0	58.7 ± 0.8	68.5 ± 9.92
	2%	✓	×	74.3 ± 0.8	87.5 ± 2.7	71.5 ± 3.5	**65.3 ± 2.1**	66.9 ± 1.7	73.1 ± 8.81
	2%	×	✓	67.5 ± 2.1	87.5 ± 2.6	69.3 ± 2.5	63.0 ± 1.7	62.6 ± 1.0	70.0 ± 10.2
	2%	✓	✓	**78.5 ± 1.5**	**87.5 ± 0.9**	73.5 ± 2.0	64.8 ± 0.3	**67.2 ± 1.1**	**74.3 ± 9.13**

TABLE II: Overview of the performance of MDT and baselines with and without our proposed Self-Supervised Losses on several LIBERO Task suites. All results show the average performance of all tasks averaged over 20 rollouts each and with 3 seeds. MDT does outperform the Transformer-BC baseline in several settings with only 2% of language annotations.

B. Evaluation on LIBERO

We further evaluate various models on LIBERO [29], a robot learning benchmark consisting of over 130 language-conditioned manipulation tasks divided into 5 different task suites: LIBERO-Spatial, LIBERO-Goal, LIBERO-Object, LIBERO-90, and LIBERO-Long. Each task suite except for LIBERO-90 consists of 10 different tasks with 50 demonstrations each. To evaluate the ability of MDT to effectively learn from partially labeled data, we only label a fraction of 2% with the associated task description. Each task suite focuses on different challenges of imitation learning: LIBERO-Goal tests on tasks with similar object categories but different goals. LIBERO-Spatial requires policies to adapt to changing spatial arrangements of the same objects. In contrast, LIBERO-Object maintains he layout while changing the objects. LIBERO-90 is the only suite that consists of 90 different tasks in several diverse environments and tasks with various spatial layouts. During evaluation, we test all models on all tasks with 20 rollouts each and average the results over 3 seeds. During the experiments, we restrict all policies to only use a static camera and a wrist-mounted one. Further details for all task suites are provided in subsection E of the Appendix.

Baselines. For our experiments in LIBERO, we report the performance of MDT, Distill-D and the best transformer baseline policy from the original benchmark, which was trained with full language annotations [29].

Results. In the LIBERO task suites, summarized in Table II, MDT proves to be effective with sparsely labeled data, outperforming the Oracle-BC baseline, which relies on fully labeled demonstrations. MDT not only outperforms the fully language-labeled Transformer Baseline in three out of four challenges but also significantly surpasses the U-Net-based Distill-D policy in all tests by a wide margin, even without auxiliary objectives. The proposed auxiliary objectives further improve the average performance of MDT by 8.5% averaged over all 5 task suites. These outcomes highlight the robustness of our architecture and affirmatively answer Question (I) regarding its efficiency.

C. Real Robot Experiments

We investigate research question (III) by assessing the ability of MDT to learn language-guided manipulations from partially labeled data in a real-world setting.

Robot Setup. MDT is evaluated on a real-world play kitchen setup with a 7 DoF Franka Emika Panda Robot. The toy kitchen has an oven, a microwave, a cooler and a sink. In addition, we positioned a toaster, a pot and a banana in the environment for the robot to interact with. A detailed overview of our setup is given in Figure 6 in the Appendix. The setup incorporates two static RGB cameras: one positioned above the kitchen for a bird's-eye view, and another placed on the right side of the robot. The action space is the normalized joint space, $[-1, 1]$, of the robot and the binary gripper control.

Play Dataset. The real-world play dataset encompasses around 4.5 hours of interactive play data with 20 different tasks for the policies to learn. Long-horizon demonstrations consisting of several tasks have been collected by volunteers via teleoperation. The volunteers were not given any instructions on how many tasks their demonstration should contain, which tasks they should perform, in which order tasks should be performed or which object they should interact with. The resulting demonstrations vary greatly in their duration and hence the number of contained tasks. The demonstrations last from around 30 seconds to more than 450 seconds, and contain between 5 and 20 tasks. The dataset is partially labeled by randomly identifying some tasks in the demonstrations and annotating the respective interval, yielding a total of 360 labels or approximately 20% of the dataset. Stationary states at the beginning and end of each demonstration were trimmed and the camera view was cropped to exclude the teleoperator from the images. Other than these adjustments no additional preprocessing was performed on the demonstrations. Hence, the methods have to learn multimodal goal-conditioned policies from partially labeled, unsegmented, long-horizon play data. Training a single agent to perform 20 different skills from such a dataset is a very hard challenge for the tested approaches. More detailed descriptions of all 20 tasks with additional visualizations are provided in Figure 7 in the Appendix.

Single Task Evaluation. First, we test several policies to complete various single tasks from the play dataset. We test all policies on a single task setting with 5 rollouts per task. During each rollout the robot starts from a central, randomized starting position, which was not used in training. Human observers decide if a task was solved successfully for each rollout. We compare MDT with our proposed auxiliary objectives against MDT trained without them and against MT-ACT. The goals were given as language instructions. The results are

Model	Avg. Single Task
MT-ACT	0.25 ± 0.43
MDT	0.51 ± 0.50
MDT + $\mathcal{L}_{\text{MGF}}$ + $\mathcal{L}_{\text{CLA}}$	$\mathbf{0.58 \pm 0.49}$

TABLE III: The single task performance on the real world dataset with language-conditioned goals. The average is computed over 5 rollouts for each of the 20 tasks. The poor performance of MT-ACT showcases the difficulty of this experiment. MDT performs significantly better with an additional boost through the auxiliary objectives. and MDT + auxiliary losses.

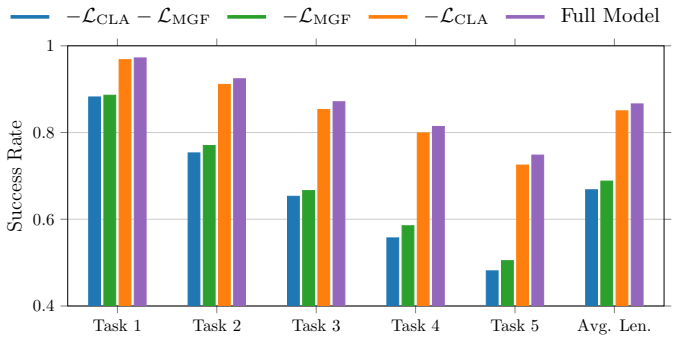


Fig. 4: Study on the performance of our proposed Masked Generative Foresight Loss and the Contrastive Latent Alignment Loss for our proposed MDT policy. We analyse the impact of both auxiliary tasks on the ABCD CALVIN challenge. The results show the average rollout length over 1000 instruction chains averaged over 3 seeds.

averaged over 5 rollouts for each of the 20 tasks. The results are summarized in Table III. The poor performance of 0.25 success rate for a strong state-of-the-art baseline such as MT-ACT highlights how challenging this setting is. In comparison, MDT achieves a respectable 0.51 success rate without and 0.58 with the auxiliary objectives. This improvement is consistent with the evaluations on the LIBERO benchmark.

Long-Horizon Multi-Task Evaluation. Finally, we test the approaches to complete several instructions in a single sequence. This requires policies to chain different behaviors together. An example instruction chain is: "Push toaster", "Pickup toast and put it to sink", "Move banana from right stove to sink", "Move pot from left to right stove", "Open oven", "Open microwave". During a rollout, the model only gets the next goal, if it has completed the prior task successfully. We test all policies using 4 different instruction chains. These instruction chains are detailed in the Appendix. We evaluate language and image goal-conditioning and report the average rollout length over all chains averaged over 4 rollouts each. The results are summarized in Table IV. Given the increased complexity of this experiment over the single task evaluation the modest performance of all three methods is not surprising, further highlighting the difficulty of this setting. Especially when considering, that the entire dataset only encompasses $4.5h$ of unsegmented play-data. MT-ACT again performs significantly worse than MDT, with an average rollout length of 0.81 vs 1.38 and 0.13 vs 0.56 for the language and image goals respectively. Moreover, the experiment shows an even stronger contribution of the proposed auxiliary objectives to the overall performance compared to the single task evaluation. Further details are discussed in the Appendix.

The promising results of MDT compared to the state-of-the-art MT-ACT baseline in this challenging setting strongly suggests an affirmative answer to Question (III), especially with the proposed auxiliary CLA and MGF objectives.

D. Evaluation of Auxiliary Losses

Next, we investigate the significance of our auxiliary self-supervised loss functions, specifically the CLA and MGF loss, on MDT's performance. Figure 4 shows the performance metrics of the ablated versions with and without these losses. The inclusion of MGF notably enhances MDT's performance on the CALVIN ABCD→D benchmark, improving average rollout lengths by over 25%. Detailed results supporting the essential role of these auxiliary tasks in MDT-V are presented in Table IX within the Appendix, showing that MDT-V surpasses all baselines with an average rollout length of 4.12 even in the absence of these two losses.

We further study the impact of MGF and CLA on the LIBERO benchmark (summarized in Table II), where the auxiliary objectives improve MDT's success rates in 4 out of 5 task suites, achieving more than a 8.5% increase on average. The results of these experiments are summarized in Table II. Interestingly, we observe a synergistic effect when both losses are applied together. Overall, on LIBERO the performance impact of CLA is higher than that of MGF, especially on the LIBERO-90 benchmark. A high number of different task labels, as it is the case for LIBERO-90, automatically implies a higher number of contrastive labels for any given tasks. We hypothesize that this increased number of contrastive labels leads to the higher performance impact of CLA for LIBERO-90. The impact on the LIBERO-Goal is the lowest, given the high initial task success rate. However, the LIBERO-Long benchmark does not seem to benefit from either MGF or CLA. The demonstrations of the LIBERO-Long benchmark consist of several sub-tasks each with a single high-level description for the entire task. We assume that this lack of sub-goals prevents the auxiliary losses from providing notable benefits.

To investigate if MGF provides a generally beneficial auxiliary objective we integrate it with MT-ACT and evaluate the model for the full CALVIN ABCD→D benchmark, as detailed in Table XI in the Appendix. MGF significantly boosts MT-ACT's average CALVIN performance by 44%, without any other modifications to the model or its hyperparameters. Similarly to the MDT results, MGF also enhances the performance of MT-ACT to learn better from multimodal goals with few language annotations. These positive outcomes for MDT, along with its effective application to other transformer-based policies, positively answer research questions (IIa) and (IIb).

Goal Modality	Model	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Avg. Rollout Length
Language	MT-ACT	50%	18.75%	12.50%	0%	0%	0%	0.81 ± 1.01
	MDT	75%	43.75%	18.75%	0%	0%	0%	1.38 ± 1.05
	MDT + $\mathcal{L}_{\text{MGF}}$ + $\mathcal{L}_{\text{CLA}}$	81.25%	56.25%	12.50%	6.25%	0%	0%	**1.56 ± 1.06**
Images	MT-ACT	12.50%	0%	0%	0%	0%	0%	0.13 ± 0.33
	MDT	12.50%	12.50%	6.25%	6.25%	0%	0%	0.38 ± 1.05
	MDT + $\mathcal{L}_{\text{MGF}}$ + $\mathcal{L}_{\text{CLA}}$	37.50%	6.25%	6.25%	6.25%	0%	0%	**0.56 ± 1.00**

TABLE IV: The average rollout length of the different approaches evaluated on the challenging long-horizon real robot play kitchen dataset. The performance is averaged over 4 instruction chains with 4 rollouts each. MDT clearly outperforms MT-ACT. The performance of MDT is further increased substantially by the auxiliary CLA and MGF objectives. The relatively short avg rollout lengths emphasize how challenging this setting is, even for strong state-of-the-art methods such as MT-ACT.

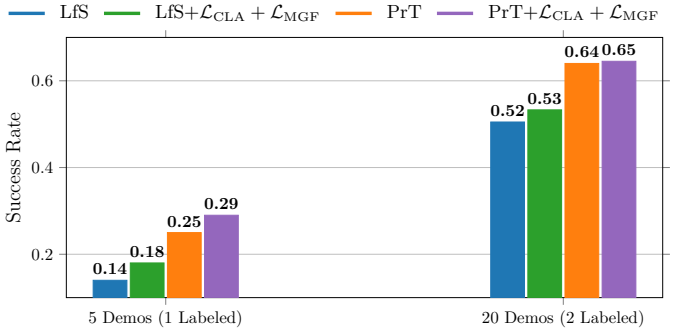


Fig. 5: Study on the performance of our proposed MGF and CLA objectives for pretraining on action-free data. We pretrain MDT on LIBERO-90 with the objectives and test the average performance on all LIBERO-Long tasks with different number of demonstrations. The results show the success rate averaged over 20 rollouts for all 10 tasks and 3 seeds. LfS refers to trained from scratch and PrT are all pretrained models.

E. Additional Ablation Studies

This section investigates several design choices for MDT and the proposed auxiliary objectives. First, we explore the importance of using pre-trained CLIP embeddings and test MGF and CLA as pretraining objective for diffusion policies. Next, we ablate several design decisions of MGF and our proposed transformer architecture and how they impact performance.

Choice of Goal Image Encoder. We analyze the importance of using pre-trained CLIP encoders for MDT to process goal images. MDT is tested with ResNet Encoders for goal images, which are trained from scratch together with text embedding of a frozen CLIP model on CALVIN ABCD. The MDT model without our auxiliary tasks achieves an average rollout length of 3.34, which is equal to the MDT variant with the frozen CLIP embeddings. The MDT variant with both auxiliary objectives achieves an average rollout length of 4.31, which is slightly worse compared to the variant with pre-trained encoders average performance of 4.41. The detailed results of this experiment are summarized in Table IX of the Appendix. Overall, the ablation shows that pretrained CLIP image embeddings are not required for MDT to succeed in learning from partially labeled datasets.

Pretraining with MGF and CLA. In this analysis, we assess the capability of CLA and MGF for pretraining MDT on only video-data without access to the robot actions. Therefore, we pretrain MDT with both auxiliary tasks on LIBERO-90 and test its downstream performance on LIBERO-Long, which contains unseen tasks. We compare the pretrained variant against MDT trained from scratch on this dataset and ablations, that omit auxiliary objectives for fine-tuning. For the LIBERO-Long experiment, we restrict all policies to learn from 5 or 20 demonstrations only. For both settings we only label 10% of all trajectories with the associated text instruction. The results of these experiments are summarized in Figure 5. Initializing MDT with our pretrained weights boost the performance in the 5 demonstration setting by 100% and by 25% in the variant with 20 demonstrations. Further, these experiments demonstrate, that both auxiliary objectives also improve performance of MDT with fewer demonstrations on LIBERO-Long independent on the pretraining. The performance increase of MDT trained from scratch with both auxiliary objectives also verifies, that CLA and MGF help MDT to learn better in scenarios with small datasets.

Masked Generative Foresight. Next, we study the different design choices of our MGF loss and compare them against ablations. Our primary focus is on assessing the impact of different masking ratios, ranging from 0.5 to 1, where 1 corresponds to a full reconstruction of the initial future image. The results indicate that a masking ratio of 0.75 achieves the best average performance, which is a value commonly used in other masking methods [22]. Thus, we use it as the default masking rate across all experiments in the paper. Further details of this analysis are provided in Table VIII in the Appendix. Additionally, we investigate the ideal foresight distance for MGF and evaluate it in two environments. MDT adopts a foresight distance of $v = 3$ as this setting consistently delivers strong performance across various scenarios. While a higher foresight distance of $v = 9$ does exhibit similar performance to a short distance of $v = 1$, it is also associated with increased variance in results. Further results of these investigations are presented in Table X in the Appendix.

Transformer Architecture. MDT is tested against two Diffusion Transformer architectures previously described in [6]. The ablations are visualized in Figure 8 of the Appendix. These comparisons are conducted on the CALVIN ABCD→D challenge, with detailed results featured in IX in the Appendix.

In the first ablation study, we incorporated a noise token as an additional input to the transformer encoder. This was done to assess the effect of excluding adaLN noise conditioning. The second ablation represents the diffusion transformer architecture from [6], which does not use any encoder module. MDT-V, when trained without any auxiliary objective, achieves an average rollout length of 4.18. The ablation without adaLN conditioning only achieves an average rollout length of 3.58. Notably, the complete omission of the transformer encoder led to a significantly lower average rollout length of 1.41. The experiments show, that the additional transformer encoder is crucial for diffusion policies to succeed in learning from different goals. In addition, separating the denoising process from the encoder and using adaLN conditioning further helps to boost performance and efficiency.

F. Limitations

While MDT shows strong performance on learning from multimodal goals, it still has several limitations: 1) While we verify the effectiveness of our method in many tasks, MGF and CLA do not increase the performance on LIBERO-Long, 2) The performance impact of MGF and CLA varies across different benchmarks. On LIBERO, CLA has a higher impact, and on CALVIN, it's the opposite. 3) Diffusion Policies require multiple forward passes to generate an action sequence, resulting in lower inference speed compared to non-diffusion approaches. 4) The average rollout lengths of MDT and the baseline on the real robot multi-tasks are relatively short. We credit this to the difficulty of the setting itself. Learning from partially labeled, unsegmented, long-horizon play data is a very challenging task. We further hypothesize that the placement of the cameras is not ideal, as the robot significantly suffers from self-occlusion. The introduction on an in-hand camera could alleviate this problem.

V. Conclusion

In this work, we introduce MDT, a novel continuous-time diffusion policy adept at learning long-horizon manipulation from play, requiring as little as 2% language labels for effective training. To further improve effectiveness, we propose MGF and CLA as simple, yet highly effective auxiliary objectives to learn more expressive behavior from multimodal goal specifications. By reconstructing future states from multimodal goal specification and aligning these state representations in the latent space, the auxiliary objectives improve downstream policy learning without additional cost during inference. We rigorously tested MDT across a diverse set of 184 tasks in both simulated environments and real-world settings. These extensive experiments not only validate our proposed auxiliary loss but also demonstrate the efficiency of the MDT policy. Notably, MDT sets two records on the CALVIN benchmark and improves over prior sota with an average 15% absolute performance increase. Moreover, in detailed studies, we demonstrate that our auxiliary objectives improve learning from multimodal goals.

In future work, we would like to investigate the advantages of additional goal-modalities like sketches in MDT. Furthermore, we plan to scale MDT towards a versatile foundation policy by pretraining the model on the large-scale, partially labeled Open-X-Embodiment dataset [7].

Acknowledgments

The work was funded by the German Research Foundation (DFG) – 448648559. The authors also acknowledge support by the state of Baden-Württemberg through HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the German Federal Ministry of Education and Research.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736, 2022.

[2] Philipp Becker, Sebastian Markgraf, Fabian Otto, and Gerhard Neumann. Reinforcement learning from multiple sensors via joint representations. *arXiv preprint arXiv:2302.05342*, 2023.

[3] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking, 2023.

[4] Denis Blessing, Onur Celik, Xiaogang Jia, Moritz Reuss, Maximilian Xiling Li, Rudolf Lioutikov, and Gerhard Neumann. Information maximizing curriculum: A curriculum-based approach for learning versatile skills. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=7eW6NzSE4g.

[5] Lili Chen, Shikhar Bahl, and Deepak Pathak. Playfusion: Skill acquisition via diffusion from language-annotated play. In *7th Annual Conference on Robot Learning*, 2023.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[7] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[8] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=c7rM7F7jQjN.

[9] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid,

Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[10] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *arXiv preprint arXiv:2302.00111*, 2023.

[11] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[12] Xinyang Geng, Hao Liu, Lisa Lee, Dale Schuurmans, Sergey Levine, and Pieter Abbeel. Multimodal masked autoencoders learn transferable representations. *arXiv preprint arXiv:2205.14204*, 2022.

[13] Nikolaos Gkanatsios, Ayush Jain, Zhou Xian, Yunchu Zhang, Christopher Atkeson, and Katerina Fragkiadaki. Energy-based Models are Zero-Shot Planners for Compositional Scene Rearrangement. In *Robotics: Science and Systems*, 2023.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284, 2020.

[15] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, Priya Sundaresan, Peng Xu, Hao Su, Karol Hausman, Chelsea Finn, Quan Vuong, and Ted Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *International Conference on Learning Representations*, 2024.

[16] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *7th Annual Conference on Robot Learning*, 2023.

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[19] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision*, pages 417–433. Springer, 2022.

[20] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.

[21] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. In *Fortieth International Conference on Machine Learning*, 2023.

[22] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[24] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to Act from Actionless Video through Dense Correspondences. *arXiv:2310.08576*, 2023.

[25] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[26] Xiang Li, Varun Belagali, Jinghuan Shang, and Michael S Ryoo. Crossway diffusion: Improving diffusion-based visuomotor policy via self-supervised learning. *arXiv preprint arXiv:2307.01849*, 2023.

[27] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, 2024.

[28] Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. *arXiv preprint arXiv:2306.00937*, 2023.

[29] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[30] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[31] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

[32] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[33] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[34] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman,

Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2022.

[35] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. URL https://openreview.net/forum?id=NJtSbIWmt2T.

[36] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):11205–11212, 2022.

[37] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022.

[38] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11576–11582. IEEE, 2023.

[39] Vivek Myers, Andre Wang He, Kuan Fang, Homer Rich Walke, Philippe Hansen-Estruch, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal representations for instruction following: A semi-supervised language interface to control. In *7th Annual Conference on Robot Learning*, 2023.

[40] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[41] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation, 2021.

[42] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[44] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

[45] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=wMpOMO0Ss7a.

[46] Krishan Rana, Andrew Melnik, and Niko Sünderhauf. Contrastive language, action, and state pre-training for robot learning. *arXiv preprint arXiv:2304.10782*, 2023.

[47] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal conditioned imitation learning using score-based diffusion policies. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[48] Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task-agnostic offline reinforcement learning. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=ViYLaruFwN3.

[49] Paul Maria Scheikl, Nicolas Schreiber, Christoph Haas, Niklas Freymuth, Gerhard Neumann, Rudolf Lioutikov, and Franziska Mathis-Ullrich. Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects. *arXiv preprint arXiv:2312.10008*, 2023.

[50] Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. *arXiv preprint arXiv:2302.02408*, 2023.

[51] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=agTr-vRQsa.

[52] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

[53] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. In *Conference on Robot Learning*, pages 2663–2682. PMLR, 2023.

[54] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[55] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[56] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[57] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[58] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances*

in Neural Information Processing Systems, 32, 2019.

[59] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

[60] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[61] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, Chelsea Finn, and Karol Hausman. Open-world object manipulation using pre-trained vision-language model. In *arXiv preprint*, 2023.

[62] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7): 1661–1674, 2011. doi: 10.1162/NECO_a_00142.

[63] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=hRZ1YjDZmTo.

[64] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024.

[65] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, and Katerina Fragkiadaki. Unifying diffusion models with action detection transformers for multi-task robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023.

[66] Ted Xiao, Harris Chan, Pierre Sermanet, Ayzaan Wahid, Anthony Brohan, Karol Hausman, Sergey Levine, and Jonathan Tompson. Robotic skill acquisition via instruction augmentation with vision-language models. *arXiv preprint arXiv:2211.11736*, 2022.

[67] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[68] Albert Zhan, Ruihan Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin. Learning visual robotic control efficiently with contrastive pre-training and data augmentation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4040–4047. IEEE, 2022.

[69] Edwin Zhang, Yujie Lu, William Wang, and Amy Zhang. Language control diffusion: Efficiently scaling through space, time, and tasks. In *International Conference on Learning Representations*, 2024.

[70] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[71] Hongkuan Zhou, Zhenshan Bing, Xiangtong Yao, Xiaojie Su, Chenguang Yang, Kai Huang, and Alois Knoll. Language-conditioned imitation learning with base skill priors under unstructured data. *arXiv preprint arXiv:2305.19075*, 2023.

[72] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R Sanketi, Grecia Salazar, Michael S Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J Joshi, Alex Irpan, brian ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=XMQgwiJ7KSX.

Hyperparameter	CALVIN	LIBERO	Real World
Number of Layers	6	2	2
Hidden Dimension	192	192	192
Image resolution	112	112	112
Masking Ratio	0.75	0.75	0.75
MLP Ration	4	4	4
Patch size	16	16	16
Norm Pixel Loss	True	True	True

TABLE V: Overview of the chosen hyperparameters for our Image Demasking Model used in the MGF loss, that uses a vision transformer architecture.

Hyperparameter	Distill-D
Action Chunk Size	8
Timestep-embed Dimensions	256
Image Encoder	ResNet18
Channel Dimensions	[512, 1024, 2048]
Learning Rate	1e-4
σ_{max}	80
σ_{min}	0.001
σ_t	0.5
Time steps	Exponential
Sampler	DDIM
Sampling Steps	10
Trainable Parameters	318 M
Optimizer	AdamW
Betas	[0.9, 0.9]
Goal Image Encoder	CLIP ViT-B/16
Goal Lang Encoder	CLIP ViT-B/32

TABLE VI: Overview of the hyperparameters for Distill-D on the CALVIN and LIBERO benchmark. Our code is based on the Diffusion-policy implementation from Chi et al. [6] with our continuous-time diffusion variant. To guarantee a fair comparison the hyperparameters for Distill-D and MDT regarding diffusion are the same.

APPENDIX

A. Diffusion Model Training and Inference

The training process for the score matching loss of MDT is summarized in Alg. 1 and the reverse diffusion process used for generating action chunks with DDIM sampler during the rollouts is summarized in Alg. 2. Further, an overview of the used hyperparameters is given in Table VII. To increase the performance, we deploy the preconditioning of Karras et al. [23]. This includes additional skip-connections and two pre-conditioning layers, which are conditioned on the current noise level σ_t for effective balancing of the high range of noise levels from 0.001 to 80

$$\begin{aligned} D_\theta(\bar{\boldsymbol{a}}_i|\boldsymbol{s}_i, \boldsymbol{g}, \sigma_t) = c_{\text{skip}}(\sigma_t)\bar{\boldsymbol{a}}_i \\ +c_{\text{out}}(\sigma_t)F_\theta(c_{\text{in}}(\sigma_t)\bar{\boldsymbol{a}}_i, \boldsymbol{s}_i, \boldsymbol{g}, c_{\text{noise}}(\sigma_t)). \end{aligned} \tag{8}$$

The utilized reconditioning functions are defined as:

- $c_{\text{skip}} = \sigma_{\text{data}}^2/(\sigma_{\text{data}}^2 + \sigma_t^2)$
- $c_{\text{out}} = \sigma_t\sigma_{\text{data}}/\sqrt{\sigma_{\text{data}}^2 + \sigma_t^2}$
- $c_{\text{in}} = 1/\sqrt{\sigma_{\text{data}}^2 + \sigma_t^2}$
- $c_{\text{noise}} = 0.25\ln(\sigma_t)$

They allow the model to decide, if it wants to predict the current noise, the denoised action sequence or something in between, depending on the current noise level [23]. For our noise distribution, we use a truncated Log-Logistic Distribution in range of $[\sigma_{\text{min}}, \sigma_{\text{max}}]$.

B. Goal Sampling Strategy

We experimented with different sampling strategies and ranges of future widows. We found that geometric sampling with a distribution probability of $p = 0.1$ works well in all tested settings on CALVIN. Other experiments with random sampling showed small drops in performance, while trials with key-state-based goal states similar to RLBench [20] did not work well in any setting. Thus, we decided to use the same strategy for CALVIN and real robot experiments.

C. Baseline Implementations

MT-ACT. An overview of the used hyperparameters of MT-ACT is given in Table VII. We tried to stay close to the recommended hyperparameters from the original paper [3] but optimized the action prediction length and the Kl-β factor for CALVIN. Empirically, we found that FiLM-conditioned ResNets do not perform well conditioned on image-goal or in combination with not using any FiLM conditioning when having image goals. Thus, we adopted default ResNets as the vision encoders for MT-ACT, as other experiments with pre-trained Voltron-Encoders [22] did not show good results and reduced the performance over 20% on the CALVIN benchmark ABCD→D.

Distill-D. We use the reported hyperparameters of Chi et al. [6] in combination with two ResNets18 and frozen CLIP encoders for visual and language goals as described in Ha et al. [16]. The resulting model contains 296.6 million trainable parameters in the 1D-CNN and an additional 22.4 million parameters for two ResNets-18. We also experimented with pretrained Voltron embeddings for Distill-D, however similar to MT-ACT Voltron did not show any performance improvements.

D. CALVIN Experiment Details

For our experiments in the CALVIN benchmark, we use the evaluation protocol as described in Mees et al. [37] for consistent comparisons with other approaches from prior work [69, 36, 27]. All methods are trained with two images from the static camera and the wrist camera. We applied random shift augmentation for both images, then resized the static camera image to 224×224 pixels and the gripper camera images to 84×84 pixels. Finally, we normalized all images with the recommended values from CLIP. The action space is delta end-effector actions and gripper signals. While Distill-D has shown a preference for position-based control, our experiments across various models demonstrated superior performance with the default setting of velocity-based control. We utilize the same CLIP image and language goal encoding models for all internally tested models. For goal-image generation in the unlabeled data segment, we employ geometric sampling with a variance of 0.1 and a future frame

Fig. 6: Overview of the real robot kitchens setup. The left image shows the play kitchen with all its objects, while the right image shows the cameras and second robot used for data collection with our robot used for teleoperation.

Hyperparameter	MT-ACT	MDT-V	MDT
Number of Encoder Layers	4	4	4
Number of Decoder Layers	6	4	6
Attention Heads	8	8	8
Action Chunk Size	10	10	10
Goal Window Sampling Size	49	49	49
Hidden Dimension	512	384	512
Action Encoder Layers	2	-	-
Action Encoder Hidden Dim	192	-	-
Latent z dim	32	-	-
Image Encoder	ResNet18	Voltron V-Cond	ResNet18
Attention Dropout	0.1	0.3	0.3
Residual Dropout	0.1	0.1	0.1
MLP Dropout	0.1	0.05	0.05
Input Dropout	0.0	0.0	0.0
Optimizer	AdamW	AdamW	AdamW
Betas	[0.9, 0.9]	[0.9, 0.9]	[0.9, 0.9]
Transformer Weight Decay	0.05	0.05	0.05
Other weight decay	0.05	0.05	0.05
Batch Size	512	512	512
Trainable Parameters	122 M	40.0 M	75.1 M
σ_{max}	-	80	80
σ_{min}	-	0.001	0.001
σ_t	-	0.5	0.5
Time steps	-	Exponential	Exponential
Sampler	-	DDIM	DDIM
Kl-β	50	-	-
Contrastive Projection	-	MAP	Single Token 1
Goal Image Encoder	CLIP ViT-B/16	CLIP ViT-B/16	CLIP ViT-B/16
Goal Lang Encoder	CLIP ViT-B/32	CLIP ViT-B/32	CLIP ViT-B/32

TABLE VII: Summary of all the Hyperparameters for the MDT policy used in the CALVIN experiments and the ones of MT-ACT.

Masking Rate	CALVIN D	LIBERO-Spatial
0.5	3.7 $\pm$ 0.04	67.8 $\pm$ 0.3
0.75	3.72 $\pm$ 0.05	67.5 $\pm$ 0.2
1	3.68 $\pm$ 0.03	63.7 $\pm$ 0.3

TABLE VIII: Ablation on different Masking Rates for Masked Generative Foresight, tested on CALVIN D→D with MDT-V and on LIBERO-Spatial with MDT.

range of $20 - 50$ to randomly select goal images for our policies. All policies are trained for twenty thousand steps on the smaller dataset and thirty thousand on the complete dataset. Extended training duration did not yield performance enhancements, and considering the substantial computational demands of training on the full dataset, we avoided prolonged training duration.

Method	$\mathcal{L}_{\text{CLA}}$	$\mathcal{L}_{\text{MGF}}$	No. Instructions in a Row (1000 chains)					
			1	2	3	4	5	**Avg. Len.**
MDT-V Abl. 1	×	×	0.914	0.782	0.675	0.588	0.487	3.58 ± 0.18
MDT-V Abl. 2	×	×	0.693	0.405	0.190	0.092	0.031	1.41 ± 0.04
MDT-V	×	×	0.971	0.907	0.840	0.766	0.698	4.18 ± 0.10
MDT-V	✓	×	0.977	0.927	0.868	0.808	0.786	4.32 ± 0.06
MDT-V	×	✓	0.986	0.946	0.903	0.851	0.794	4.48 ± 0.03
MDT-V	✓	✓	0.989	0.958	0.916	0.862	0.801	4.52 ± 0.02
MDT	×	×	0.882	0.753	0.653	0.557	0.481	3.34 ± 0.06
MDT+Goal ResNets	×	×	0.886	0.764	0.651	0.565	0.48	3.34 ± 0.05
MDT	✓	✓	0.978	0.938	0.888	0.831	0.77	4.41 ± 0.03
MDT+Goal ResNets	✓	✓	0.978	0.923	0.862	0.807	0.735	4.31 ± 0.08

TABLE IX: Overview of the performance influence of MGF and Contrastive Alignment on MDT-V on the CALVIN ABCD→D challenge. In addition, the performance of both transformer ablations are also shown. Moreover, results for MDT with ResNets as image goal encoders are reported with and without auxiliary objectives. The results are reported over 1000 rollouts averaged over 3 seeds.

	CALVIN ABCD	LIBERO-Spatial
1	4.50 ± 0.02	64.4 ± 0.4
3	**4.52 ± 0.02**	**67.5 ± 0.2**
9	4.44 ± 0.03	65.6 ± 0.5

TABLE X: Ablation on the best prediction horizon for Masked Generative Foresight, tested on CALVIN ABCD→D with MDT-V and LIBERO-Spatial with MDT.

Policy	Avg. Len. CALVIN
MT-ACT	2.80 ± 0.03
MT-ACT + $\mathcal{L}_{\text{MGF}}$	**4.03 ± 0.08**

TABLE XI: Evaluation of the Performance Increase of the MT-ACT policy with the additional Masked Generative Foresight Loss on the CALVIN ABCD→D challenge.

E. LIBERO Experiment Details

The LIBERO task suites [29] consists of 5 different ones in the benchmark with 50 demonstrations per task. To emulate a scenario with sparse language labels, we divided the dataset into two segments: one set consists of single demonstrations accompanied by language annotations, and the other comprises 49 demonstrations without labels. For generating goal images, we utilized the final state of each rollout. We used the default end-effector action space in all our experiments. Consistent with the CALVIN setup, we employed identical image augmentation methods to prepare our data. We trained all models for 50 epochs and then tested them on 20 rollouts averaged over 3 seeds. The benchmark is structured into five distinct task suites, each designed to test different aspects of robotic learning and manipulation:

- **Spatial**: This suite emphasizes the robot's ability to understand and manipulate spatial relationships. Each task involves placing a bowl, among a constant set of objects, on a plate. The challenge lies in distinguishing between two identical bowls that differ only in their spatial placement relative to other objects.
- **Goal**: The Goal suite tests the robot's proficiency in understanding and executing varied task goals. Despite using the same set of objects with fixed spatial relationships, each task in this suite differs in the ultimate goal, demanding that the robot continually adapt its motions and behaviors to meet these varying objectives.
- **Object**: Focused on object recognition and manipulation, this suite requires the robot to pick and place a unique object in each task.
- **Long**: This suite comprises tasks that necessitate long-horizon planning and execution. The Long suite is particularly challenging, as it tests the robot's ability to maintain performance and adaptability over extended task duration.
- **90**: Offering a diverse set of 90 short-horizon tasks across five varied settings.

F. Real Robot Experiments

Detailed Environment Overview. Our real robot setup with the toy kitchen is visualized in Figure 6. In the kitchen the robot can interact with the microwave positioned on the top right, the oven in the lower left half of the kitchen, the cooler on the lower-right side of the kitchen and the sink on the right side of the counter top. The robot is positioned next to a toy kitchen with the following additional objects: a banana, a pot, a toaster with toast. In total we create a set of 20 diverse tasks for the robot to learn from the partially labeled play data. All tasks of our dataset are shown in Figure 7.

Play Dataset Collection. Our volunteers collect play data with teleoperation with a leader and follower robot setup, which is visualized in Figure 6. During the teleoperation, we collect the robot's proprioceptive sensor data and two images from our two static cameras with 6 Hz. We extract the desired joint position as our action signal and normalize

Task	No.	MT-ACT	MDT	MDT MGF
Banana from rt stove to sink	1	0	0.8	1.0
Banana from sink to rt stove	2	0	0.8	1.0
Pot from rt stove to sink	3	0	0	0
Pot from sink to rt stove	4	0.4	0.8	0
Pot from lt stove to sink	5	0	0.8	0.6
Pot from sink to lt stove	6	0	0	1.0
Pot from lt to rt stove	7	0.2	0.2	0
Pot from rt to lt stove	8	0	0.6	1.0
Open microwave	9	1.0	1.0	1.0
Close microwave	10	0	1.0	0
Open oven	11	1.0	1.0	1.0
Close oven	12	0	0	0
Open ice box	13	1.0	1.0	1.0
Close ice box	14	0	1.0	0
Pull oven tray	15	0.6	0	0.2
Push oven tray	16	0.4	0	0.2
Banana from rt stove to tray	17	0	0	0.4
Banana from tray to rt stove	18	0	0	1.0
Push toaster	19	0	0.2	1.0
Toast to sink	20	0.4	1.0	1.0

TABLE XII: Detailed results of our real robot kitchen experiments for single-task setting with language goals. All results are averaged over 5 rollouts. "right" and "left" are abbreviated with "rt" and "lt".

MT-ACT (Language Goals)	T1	T2	T3	T4	T5	T6
1: Random	0.0	0.0	0.0	0.0	0.0	0.0
2: Open Close All	1.0	0.75	0.5	0.0	0.0	0.0
3: Stovetop Sink	0.5	0.0	0.0	0.0	0.0	0.0
4: Oven	0.5	0.0	0.0	0.0	0.0	-

MT-ACT (Image Goals)	T1	T2	T3	T4	T5	T6
1: Random	0.0	0.0	0.0	0.0	0.0	0.0
2: Open Close All	0.0	0.0	0.0	0.0	0.0	0.0
3: Stovetop Sink	0.5	0.0	0.0	0.0	0.0	0.0
4: Oven	0.0	0.0	0.0	0.0	0.0	-

MDT (Language Goals)	T1	T2	T3	T4	T5	T6
1: Random	0.75	0.75	0.0	0.0	0.0	0.0
2: Open Close All	1.0	1.0	0.75	0.0	0.0	0.0
3: Stovetop Sink	0.75	0.0	0.0	0.0	0.0	0.0
4: Oven	0.5	0.0	0.0	0.0	0.0	-

MDT (Image Goals)	T1	T2	T3	T4	T5	T6
1: Random	0.5	0.5	0.25	0.25	0.0	0.0
2: Open Close All	0.0	0.0	0.0	0.0	0.0	0.0
3: Stovetop Sink	0.0	0.0	0.0	0.0	0.0	0.0
4: Oven	0.0	0.0	0.0	0.0	0.0	-

MDT + $\mathcal{L}_{\text{MGF}}$ + $\mathcal{L}_{\text{LCA}}$ (Language Goals)	T1	T2	T3	T4	T5	T6
1. Random	1.0	0.75	0.25	0.25	0.0	0.0
2. Open Close All	1.0	1.0	0.25	0.0	0.0	0.0
3. Stovetop Sink	0.5	0.25	0.0	0.0	0.0	0.0
4. Oven	0.75	0.25	0.0	0.0	0.0	-

MDT + $\mathcal{L}_{\text{MGF}}$ + $\mathcal{L}_{\text{LCA}}$ (Image Goals)	T1	T2	T3	T4	T5	T6
1: Random	0.25	0.25	0.25	0.25	0.0	0.0
2: Open Close All	0.5	0.0	0.0	0.0	0.0	0.0
3: Stovetop Sink	0.25	0.0	0.0	0.0	0.0	0.0
4: Oven	0.5	0.0	0.0	0.0	0.0	-

TABLE XIII: Detailed results of our long-horizon, real robot multi-task experiments.

it in the range $[-1, 1]$. To label the data with additional text instructions, we randomly sample short sequences from our uncurated play dataset and ask a human to describe the task in this segment. In total, we generate 360 labeled short-horizon segments for the model training. For every task we query GPT-4 to generate different text instructions for more diverse language descriptions. We note, that training a single policy on such a small dataset of real world play data is very challenging for the models.

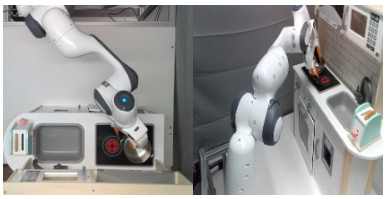
Policy Training. We train all tested policies on our processed real robot data for around 24 hours with a small cluster consisting of 4 GPUs for around 100 epochs. For checkpoint selection, we use the checkpoint with the lowest validation loss. For MT-ACT, we selected the last epoch since our prior experience with the benchmarks indicated an improvement in performance even when the validation loss began to rise again.

Evaluation Details. We collect 10 goal images of each task from our play dataset to test image-conditioning, and in addition we have a set of 10 different text instructions for each task. Each policy is tested 5 times from a starting position not seen in training with some added noise to it. Further, we test our policies on long-horizon setup, where we define 4 different instruction chains consisting of 5 or 6 tasks in sequence. During these rollouts, we observe the robot, if the policy completes the desired sub-task and only give it the next goal description if he manages to complete the prior task successfully. During our experiments, we further vary the orientation of the banana slightly for the robot to pick up, while we keep the toaster at the same position during all our experiments. Detailed results for all our experiments for single task and multi-task setting are summarized in Table XII and Table XIII. Task sequences used in the multi-task evaluation are listed in the following:

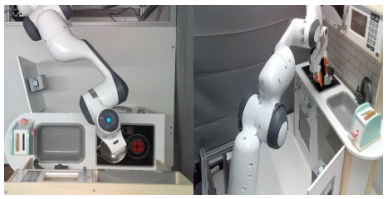
1) **Random**: Push toaster, Pickup toast and put to sink, Banana from right stove to sink, Pot from left to right stove, Open oven, Open microwave
2) **Open Close All**: Open microwave, Open oven, Open ice box, Close ice box, Close oven, Close microwave
3) **Stovetop Sink**: Banana from right stove to sink, Push toaster, Pot from left to right stove, Pickup toast and put to sink, Pot from right to left stove, Banana from sink to right stove
4) **Oven**: Open oven, Pull oven tray, Banana from right stove to oven tray, Push oven tray, Close oven

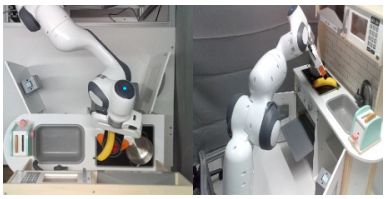
Failure Cases. Both variants of MDT struggle to solve all tasks related to moving the banana and the pot to specific positions. Especially the two tasks "Move the pot/banana from the right stove to the sink" is often misunderstood by all tested policies. We hypothesize, that the policies don't have enough labels to learn to differentiate between these similar states. Policies also struggle solving tasks where they need to close a door, or push the oven tray. These tasks are "Close oven/microwave/ice box" and "Push oven tray". Most of the time, the policies failed these tasks by a few millimeters. We hypothesize, that these tasks included tight actions which have varying degrees of "openness" and had quick demonstrations in the dataset. These factors combined is likely the reason why the policies failed these tasks by a narrow margin.


Move banana from right stove to oven tray


Move pot from sink to left stove


Open the microwave


Move pot from left to right stove


Move banana from tray to right stove


Pull the oven tray


Move banana from right stove to sink


Close the oven


Push the toaster lever


Move pot from right to left stove


Move pot from right stove to sink


Open the ice box


Open the oven


Push the oven tray


Close the microwave


Move banana from sink to right stove


Move pot from left stove to sink


Close the ice box


Pick up toast and put it in the sink

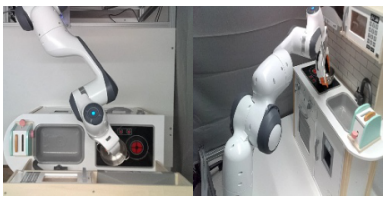

Move pot from sink to right stove

Fig. 7: Overview of the 20 tasks recorded during play from the both camera perspectives. We test our policies on these tasks during evaluation after training on a partially labeled dataset.

Algorithm 1 Score Matching Loss [58]

Require: Play Dataset $\mathcal{L}_{\text{play}}$
Require: Score Model $\mathbf{D}_\theta(\bar{\boldsymbol{a}}_i, \boldsymbol{s}_i, \boldsymbol{g}, \sigma_t)$
Require: Noise Distribution: $\text{LogLogistic}(\alpha, \beta)$
for $i \in \{0, ..., N_{\text{train steps}}\}$ **do**
  Sample $(\boldsymbol{o}, \boldsymbol{g}) \sim \mathcal{L}_{\text{play}}$
  Sample $\sigma_t \sim \text{LogLogistic}(\alpha, \beta)$
  Sample $\epsilon \sim \mathcal{N}(0, \sigma_t)$
  $\mathcal{L}_{D_\theta} \leftarrow \mathbb{E}_{\sigma, \bar{\boldsymbol{a}}_i, \boldsymbol{\epsilon}}\big[\alpha(\sigma_t) \|D_\theta(\bar{\boldsymbol{a}}_i + \boldsymbol{\epsilon}, \boldsymbol{s}_i, \boldsymbol{g}, \sigma_t) - \bar{\boldsymbol{a}}_i\|_2^2\big]$
end for

Algorithm 2 DDIM Sampler as DPM-Solver-1 [30, 57]

Require: Current state $\boldsymbol{s}_i$, goal $\boldsymbol{g}$
Require: Score Model $D_\theta(\mathbf{a}, \boldsymbol{s}_i, \boldsymbol{g}, \sigma)$
Require: Noise scheduler $\sigma_i = \sigma(t_i)$
Require: Discrete time steps $t_{i \in \{0,..,N\}}$
Draw sample $\mathbf{a}_{1:k,0} \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$
for $i \in \{0, ..., N-1\}$ **do**
  $t, t_{\text{next}} \leftarrow t_{\text{fn}}(\sigma_i), t_{\text{fn}}(\sigma_{i+1})$
  $h \leftarrow t_{\text{next}} - t$
  $\mathbf{a}_{1:k,i+1} \leftarrow \frac{\sigma_{\text{fn}}(t_{\text{next}})}{\sigma_{\text{fn}}(t)} \mathbf{a}_{1:k,i} - \exp(-h) D_\theta(\mathbf{a}_{1:k,}, \boldsymbol{s}_i, \mathbf{g}, \sigma_i)$
end for
return $\mathbf{a}_{1:k,N}$

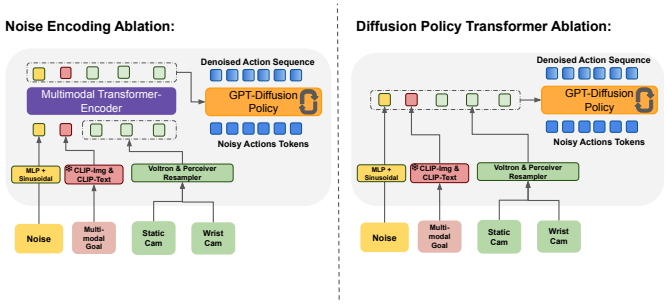


Fig. 8: Overview of the two Diffusion Transformer Baseline Architectures used for the Ablation Study. The first variant uses a transformer encoder but also processes the noise as a token. The second one is the Transformer Diffusion policy presented in Chi et al. [6] without any encoder.

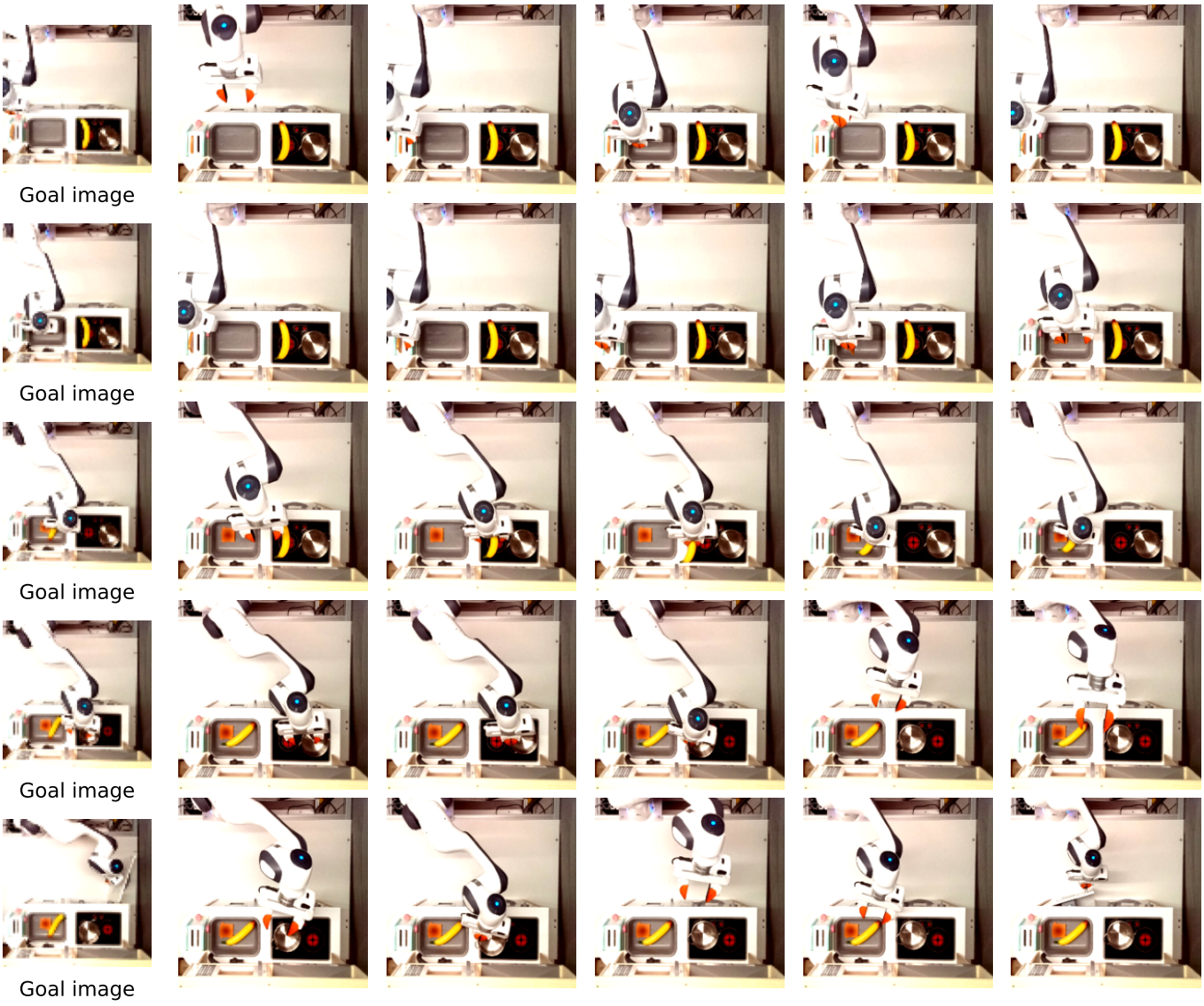


Fig. 9: Real Robot rollouts with goal image conditioning. The first column shows the goal image used for the rollout. 4 out of 6 tasks are successful. The robot fails to open the oven door and opens the ice box instead.