Multimodal Diffusion Transformer: Learning Versatile Behavior from Multimodal Goals

Moritz Reuss[1], Ömer Erdinç Yağmurlu[1], Fabian Wenzel[1], Rudolf Lioutikov[1]

***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 that enhance 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. Further, 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 demonstrated its ability to solve long-horizon manipulation from sparsely annotated data in both simulated and real-world environments.**

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 [1], [2], [3] or visual goals [4], [5]. 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 biggest 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 [1], [3], [6]. 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 [7], [8]. Unfortunately, MTIL capitalizes on large, fully annotated datasets. Collecting real human demonstrations is notably time-consuming and labor-intensive.

[1]Intuitive Robots Lab, Karlsruhe Institute of Technology, Germany

One way to circumvent these challenges is Learning from Play (LfP) [9], [10], 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 [9]. 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-RT [11], 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 [12], [5]. Diffusion Policies can learn expressive, versatile behavior conditioned on language-goals [13], [14]. Yet, none of the current methods adequately addresses the challenge of learning from multimodal goal specifications.

This work introduces a novel diffusion-based approach capable of learning versatile behavior from different goal modalities, such as language and images, simultaneously. The approach learns efficiently even if it is trained on data with few language-annotated demonstrations. The performance is further improved by introducing a simple, yet highly effective self-supervised loss, Masked Generative Foresight (MGF). This loss encourages 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 an informative latent space, which maps goals to desired future states independent of their modality. Detailed experiments and ablations show that this additional loss significantly 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 the CALVIN challenge [10], a popular benchmark for language-guided learning from play data comprised of human demonstrations with few language annotations. 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.

II. Method

MDT is a diffusion-based transformer encoder-decoder architecture that simultaneously leverages two self-supervised auxiliary objectives. Namely Contrastive Latent Alignment and Masked Generalized Foresight. First, the problem definition is provided. Next, the continuous-time diffusion formulation, essential for understanding action sequence learning from play, is discussed. Followed by an overview of the proposed transformer architecture of MDT. Afterward, the novel self-supervised Masked Generative Foresight objective and latent token alignment 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 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 [15] 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 annotation is introduced. MDT leverages a continuous time diffusion model [16], [17]. 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) [17]. MDT leverages the SDE formulation from [16]

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

commonly used in image generation [16], [18]. 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 [16], [5], [18], 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 [16]. This SDE is notable for having an associated ordinary differential equation, the Probability Flow ODE [17]. When action chunks of this ODE are sampled at time t of the diffusion process, they align with the distribution $p_t(\bar{\boldsymbol{a}}_i | \boldsymbol{s}_i, \boldsymbol{g})$,

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

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) [19]

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

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 [17]. For the experiments, MDT uses the DDIM sampler [17] to diffuse an action sequence in 10 denoising 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. An overview of the architecture is given in Figure 3 of the Appendix.

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 [20]. Each image is embedded into 196 latent tokens by Voltron. The

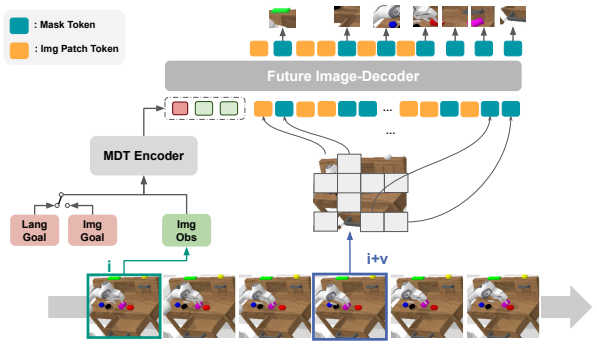

Fig. 1: The Masked Generative Foresight Auxiliary Task enhances the MDT model. It starts by encoding the current observation and goal using the MDT Encoder. The resulting latent state representations then serve as conditional inputs for the Future Image-Decoder. This decoder receives encoded patches of future camera images along with mask tokens. Its task is to reconstruct the occluded patches in future frames.

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 [12] 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 [21] 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.

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 applies AdaLN-conditioning to the Transformer Decoder blocks to condition the denoising process to the current noise level [22]. This process is illustrated in the right part of Figure 3, encapsulating all internal update steps. The proposed framework separates representation learning from denoising, which results 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 [12].

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. 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

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

where the indicator function $\mathbf{1}_{\text{mk}}(\mathbf{u})$ is 1 if $\mathbf{u}$ is masked and 0 otherwise.

MGF is conceptually simple and can be universally applied to all transformer policies. Various experiments in this work show that this auxiliary loss not only improves the behavior of MDT but also notably increases the performance of the Multi-Task Action Chunking Transformer (MT-ACT) policy [23].

E. Contrastive Alignment of 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 [21]. However, CLIP exhibits a tendency towards static images and struggles to interpret spatial relationships and dynamics as highlighted in various studies [24], [25], [26]. The 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 300 million parameter large CLIP model, MDT introduces an additional auxiliary objective that aligns the MDT(-V) embeddings across different goal modalities. These embeddings do not only include the goal but also the current state information, which allows the Contrastive Latent Alignment (CLA) objective to consider the task dynamics.

Since CLA requires a single vector for each goal modality, the various MDT latent tokens are reduced via Multihead Attention Pooling [27] and subsequently normalized. Hence, every training sample $(\boldsymbol{s}_i, \bar{\boldsymbol{a}}_i)$ that is paired with a multi-modal 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

Train→Test	Method	No. Instructions in a Row (1000 chains)					
		1	2	3	4	5	**Avg. Len.**
ABCD→D	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)	**97.5%**	**92.4%**	**87.1%**	**81.4%**	**74.8%**	**4.33±(0.08)**
	MDT-V (ours)	**98.8%**	**95.9%**	**91.2%**	**86.1%**	**79.4%**	**4.51±(0.02)**

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

$C\left(\boldsymbol{z}_i^{\mathbf{0}}, \boldsymbol{z}_i^{\mathbf{1}}\right)$ between the image and language projection

$$\mathcal{L}_{\text{CLA}} = -\frac{1}{2B}\sum_{i=1}^{B}\left(\log\left(\frac{\exp\left(\frac{C\left(\boldsymbol{z}_i^{\mathbf{0}},\boldsymbol{z}_i^{\mathbf{1}}\right)}{\upsilon}\right)}{\sum_{j=1}^{B}\exp\left(\frac{C\left(\boldsymbol{z}_i^{\mathbf{0}},\boldsymbol{z}_j^{\mathbf{1}}\right)}{\upsilon}\right)}\right) + \log\left(\frac{\exp\left(\frac{C\left(\boldsymbol{z}_i^{\mathbf{0}},\boldsymbol{z}_i^{\mathbf{1}}\right)}{\upsilon}\right)}{\sum_{j=1}^{B}\exp\left(\frac{C\left(\boldsymbol{z}_j^{\mathbf{0}},\boldsymbol{z}_i^{\mathbf{1}}\right)}{\upsilon}\right)}\right)\right), \tag{6}$$

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

III. Evaluation

This section aims to answer the following questions:

- (I) Is MDT able to learn long-horizon manipulation from play data with few language annotations?
- (IIa) Do Masked Generative Foresight and Contrastive Latent Alignment enhance the performance of MDT?
- (IIb) Does MGF improve the performance of other transformer policies?

A. Simulated Benchmark Environments

We conduct multiple simulation experiments on two popular and challenging robot learning benchmarks:

CALVIN. The CALVIN challenge [10] 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 4. 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. During the rollouts, the agent gets a reward of 1 for completing the instruction with a maximum of 5 for every rollout. We compare our proposed policy against the several state-of-the-art language-conditioned multi-task policies on CALVIN. For policies, that report results on CALVIN, we use their reported performance to guarantee a fair comparison. A detailed list of all baselines is described in Section D of the Appendix.

B. Evaluation Results

The results of our experiments on CALVIN are summarized in Table I and Table VIII. 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 4.51 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 [28] and boosting the average rollout length by 20% over the second best baseline. The results affirmatively answer Question (I).

Further, we conduct experiments on the LIBERO benchmark and a real-world play kitchen to adress the remaining questions. These experiments are described in detail in Section B of the Appendix.

IV. 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 as a simple, yet highly effective auxiliary objective to learn more expressive behavior from multimodal goal specifications. We rigorously tested MDT across a diverse set of 169 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.

REFERENCES

[1] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[2] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, "Unleashing large-scale video generative pre-training for visual robot manipulation," in *International Conference on Learning Representations*, 2024.

[3] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[4] Z. J. Cui, Y. Wang, N. M. M. Shafiullah, and L. Pinto, "From play to policy: Conditional behavior generation from uncurated robot data," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=c7rM7F7jQjN

[5] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal conditioned imitation learning using score-based diffusion policies," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[6] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, brian ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=XMQgwiJ7KSX

[7] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," *arXiv preprint arXiv:2005.07648*, 2020.

[8] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters*, 2023.

[9] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," in *Conference on robot learning*. PMLR, 2020, pp. 1113–1132.

[10] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, 2022.

[11] O. X.-E. Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu, C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns, F. Ceola, F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, H. Su, H.-S. Fang, H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Kim, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Wu, J. Luo, J. Gu, J. Tan, J. Oh, J. Malik, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Zhang, K. Majd, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Tan, L. Ott, L. Lee, M. Tomizuka, M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundaresan, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Moore, S. Bahl, S. Dass, S. Song, S. Xu, S. Haldar, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. hua Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Xu, and Z. J. Cui, "Open X-Embodiment: Robotic learning datasets and RT-X models," https://arxiv.org/abs/2310.08864, 2023.

[12] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[13] Z. Xian, N. Gkanatsios, T. Gervet, and K. Fragkiadaki, "Unifying diffusion models with action detection transformers for multi-task robotic manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[14] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *7th Annual Conference on Robot Learning*, 2023.

[15] D. Blessing, O. Celik, X. Jia, M. Reuss, M. X. Li, R. Lioutikov, and G. Neumann, "Information maximizing curriculum: A curriculum-based approach for learning versatile skills," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=7eW6NzSE4g

[16] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[17] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2020.

[18] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," *arXiv preprint arXiv:2303.01469*, 2023.

[19] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[20] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 716–23 736, 2022.

[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[22] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[23] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," 2023.

[24] T. Xiao, H. Chan, P. Sermanet, A. Wahid, A. Brohan, K. Hausman, S. Levine, and J. Tompson, "Robotic skill acquisition via instruction augmentation with vision-language models," *arXiv preprint arXiv:2211.11736*, 2022.

[25] V. Myers, A. W. He, K. Fang, H. R. Walke, P. Hansen-Estruch, A. Kolobov, A. Dragan, and S. Levine, "Goal representations for instruction following: A semi-supervised language interface to control," in *7th Annual Conference on Robot Learning*, 2023.

[26] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 11 205–11 212, 2022.

[27] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang, "Language-driven representation learning for robotics," *arXiv preprint arXiv:2302.12766*, 2023.

[28] E. Zhang, Y. Lu, W. Wang, and A. Zhang, "Language control diffusion: Efficiently scaling through space, time, and tasks," in *International Conference on Learning Representations*, 2024.

[29] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 576–11 582.

[30] L. Chen, S. Bahl, and D. Pathak, "Playfusion: Skill acquisition via diffusion from language-annotated play," in *7th Annual Conference on Robot Learning*, 2023.

[31] N. Gkanatsios, A. Jain, Z. Xian, Y. Zhang, C. Atkeson, and K. Fragkiadaki, "Energy-based Models are Zero-Shot Planners for Compositional Scene Rearrangement," in *Robotics: Science and Systems*, 2023.

[32] A. Jain, N. Gkanatsios, I. Mediratta, and K. Fragkiadaki, "Bottom up top down detection transformers for language grounding in images and point clouds," in *European Conference on Computer Vision*. Springer, 2022, pp. 417–433.

[33] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning," in *7th*

Annual Conference on Robot Learning, 2023. [Online]. Available: https://openreview.net/forum?id=wMpOMO0Ss7a

[34] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, C. Finn, and K. Hausman, "Open-world object manipulation using pre-trained vision-language model," in *arXiv preprint*, 2023.

[35] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[36] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[37] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu *et al.*, "Vision-language foundation models as effective robot imitators," in *International Conference on Learning Representations*, 2024.

[38] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard, "Latent plans for task-agnostic offline reinforcement learning," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=ViYLaruFwN3

[39] H. Zhou, Z. Bing, X. Yao, X. Su, C. Yang, K. Huang, and A. Knoll, "Language-conditioned imitation learning with base skill priors under unstructured data," *arXiv preprint arXiv:2305.19075*, 2023.

[40] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=hRZ1YjDZmTo

[41] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," in *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. [Online]. Available: https://openreview.net/forum?id=agTr-vRQsa

[42] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, and F. Mathis-Ullrich, "Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects," *arXiv preprint arXiv:2312.10008*, 2023.

[43] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[44] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[45] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[46] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *arXiv preprint arXiv:2302.00111*, 2023.

[47] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum, "Learning to Act from Actionless Video through Dense Correspondences," *arXiv:2310.08576*, 2023.

[48] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[49] X. Geng, H. Liu, L. Lee, D. Schuurmans, S. Levine, and P. Abbeel, "Multimodal masked autoencoders learn transferable representations," *arXiv preprint arXiv:2205.14204*, 2022.

[50] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.

[51] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel, "Multi-view masked world models for visual robotic manipulation," *arXiv preprint arXiv:2302.02408*, 2023.

[52] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. 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. [Online]. Available: https://openreview.net/forum?id=NJtSbIWmt2T

[53] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[54] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," 2021.

[55] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[56] A. Zhan, R. Zhao, L. Pinto, P. Abbeel, and M. 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)*. IEEE, 2022, pp. 4040–4047.

[57] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[58] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5639–5650.

[59] P. Becker, S. Markgraf, F. Otto, and G. Neumann, "Reinforcement learning from multiple sensors via joint representations," *arXiv preprint arXiv:2302.05342*, 2023.

[60] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," in *The Eleventh International Conference on Learning Representations*, 2022.

[61] K. Rana, A. Melnik, and N. Sünderhauf, "Contrastive language, action, and state pre-training for robot learning," *arXiv preprint arXiv:2304.10782*, 2023.

[62] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning*. PMLR, 2023, pp. 416–426.

[63] X. Li, V. Belagali, J. Shang, and M. S. Ryoo, "Crossway diffusion: Improving diffusion-based visuomotor policy via self-supervised learning," *arXiv preprint arXiv:2307.01849*, 2023.

[64] R. Shah, R. Martín-Martín, and Y. Zhu, "Mutex: Learning unified policies from multimodal task specifications," in *Conference on Robot Learning*. PMLR, 2023, pp. 2663–2682.

[65] S. Lifshitz, K. Paster, H. Chan, J. Ba, and S. McIlraith, "Steve-1: A generative model for text-to-behavior in minecraft," *arXiv preprint arXiv:2306.00937*, 2023.

[66] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," in *Fortieth International Conference on Machine Learning*, 2023.

[67] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, P. Sundaresan, P. Xu, H. Su, K. Hausman, C. Finn, Q. Vuong, and T. Xiao, "Rt-trajectory: Robotic task generalization via hindsight trajectory sketches," in *International Conference on Learning Representations*, 2024.

[68] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *arXiv preprint arXiv:2306.03310*, 2023.

APPENDIX

A. 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 [3], [29], [23], [26], [8], [30], [13] and improved language expression-grounding [31], [32], [33], [24], [34]. Notably, methods like CLIPPort [35] employ frozen CLIP embeddings for language-guided pick and place, while others, such as PaLM-E [36] and RoboFlamingo [37], finetune vision-language models for robot control. Other methods focus on hierarchical skill learning for language-guided manipulation in LfP [10], [9], [29], [38], [39], [40]. Further transformer-based methods without hierarchical structures [6], [4], [5], [2], [41], [42], focus on next-action prediction based on previous observation tokens. 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 [23], [43], [44].

Furthermore, diffusion-based methods have gained adoption as policy representations that iteratively diffuse actions from Gaussian Noise [45], [17]. Several diffusion policy approaches focus on generating plans on different abstraction levels for behavior generation. LAD [28] trains a diffusion model to diffuse a latent plan sequence in the pre-trained latent space of HULC [26] combined with HULC's low-level policy. UniPy [46] and AVDC [47] directly plan in the image space using a video diffusion model and execute the plan with another model. Frameworks related to our approach are Distill-Down [14] and Play-Fusion [30], which also utilize a Diffusion Policy for language-guided policy learning. Both methods use a variant of the CNN-based diffusion policy [12]. However, all these methods require fully annotated datasets to learn language-conditioned policies. MDT effectively learns from multimodal goals, which enables 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 [27], [48], [49], [50], [51], [52] or contrastive objectives [53], [54], [55], [56], [57], [58], [59], [60], [61]. Another body of work explores robust representations for robot policies from multiple sensors, using token masking strategies [62] or generative video generation [2]. 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 [63] proposes a self-supervised loss specifically designed for CNN-based diffusion policies [12] 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.

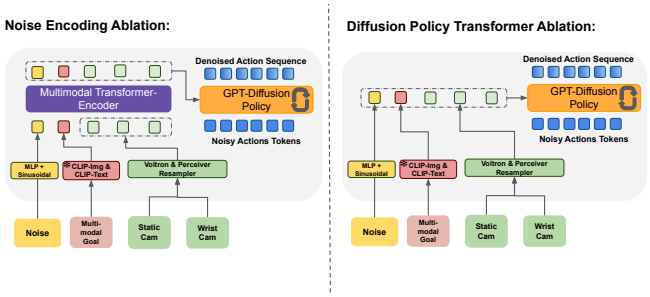

Fig. 2: Overview of the two Diffusion Transformer Baseline Architectures used for the Ablation Study. The first variant uses a transformer encoder but also processes the noise as a token. The second one is the Transformer Diffusion policy [12] without any encoder.

Recent trends show an increase in transformer-based policies that encode only the current state information without any history of prior states [23], [43], [44] to predict a sequence of future actions in an efficient manner. Traditional token masking strategies or video generation objectives that rely on token sequences of multiple states are incompatible with these models due to their unique operational frameworks. To bridge this gap, our proposed MGF objective is tailored to enhance the capabilities of these single-state observation policies. MGF enables the learning of versatile behavior from multimodal goals efficiently and without additional inference costs.

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 [64] 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 [65] 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 [51] or employing multimodal prompts [66] that combine text with image descriptions. Rt-Sktech and Rt-Trajectory present two new conditioning methods leveraging goal sketches of the desired scene [6] and sketched trajectories of the desired motion [67], respectively. While our method 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.

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 II: Overview of the chosen hyperparameters for our Image Demasking Model used in the Masked-Generative Foresight loss, that consists of 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
$\sigma_{\max}$	80
$\sigma_{\min}$	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 III: Overview of the hyperparameters for Distill-D on the CALVIN and LIBERO benchmark. Our code is based on the Diffusion-policy implementation [12] with our continuous-time diffusion variant. To guarantee a fair comparison the hyperparameters for Distill-D and MDT Diffusion are chosen.

B. Additional Experiments

In the following section, we describe our experiments on the LIBERO benchmark and a real robot play kitchen in detail.

C. LIBERO Experiment Details

The LIBERO task suites [68] 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 durations.
- **90**: Offering a diverse set of 90 short-horizon tasks across five varied settings.

D. Baselines

We compare MDT against several state-of-the-art policies, described in detail below:

- **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 [26].
- **LAD:** A hierarchical diffusion policy, that extends the HULC policy by substituting the high-level planner with a U-Net Diffusion model [28] to diffuse plans.
- **Distill-D:** A language-guided Diffusion policy from [14], that extends the initial U-Net diffusion policy [12] 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 [23], [44], 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 [37] 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. Further, we directly compare the self-reported results of HULC, LAD, and RoboFlamingo on CALVIN [28], [26], [37], [28]. All models use the same language and image goal models to further ensure fair comparisons. Since RoboFlamingo only published the best seed of each model, we can not include standard deviations in their results. 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 [68]. During the

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

Masking Rate	CALVIN ABCD	LIBERO-Spatial
0.5	3.54 $\pm$ 0.04	67.8 $\pm$ 0.3
0.75	3.60 $\pm$ 0.05	67.5 $\pm$ 0.2
1	3.50 $\pm$ 0.03	63.7 $\pm$ 0.3

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

experiments, we restrict all policies to only use a static camera and a wrist-mounted one.

LIBERO. We evaluate various models on LIBERO [68], a robot learning benchmark consisting of over 130 language-conditioned manipulation tasks divided into 5 different task suites, with different focus. Details are provided in Section C of the Appendix. Every task in each suite has 50 demonstrations, where we only label one demonstration with the associated task description and all others without. During evaluation, we test all models on all tasks with 20 rollouts each and average the results over 3 seeds.

E. Evaluation on LIBERO

In the LIBERO task suites, summarized in Table IX, MDT proves to be effective with sparsely labeled data, outperforming the Oracle-BC baseline, which relies on fully labeled demonstrations from LIBERO. 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, even without auxiliary objectives.

F. Evaluation of Masked Generative Foresight

We next investigate the significance of our auxiliary self-supervised loss functions, specifically the CLA and MGF loss, on MDT's performance. Figure 5 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 VI 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 IX), where the auxiliary objectives improve MDT's success rates in 4 out of 5 test suites, achieving more than a 5.4% increase on average. The results of these experiments are summarized in Table IX. Interestingly, we observe a synergistic effect when both losses are applied together. 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 hypothesize 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. MGF significantly boosts MT-ACT's average CALVIN performance by 44%, without any other

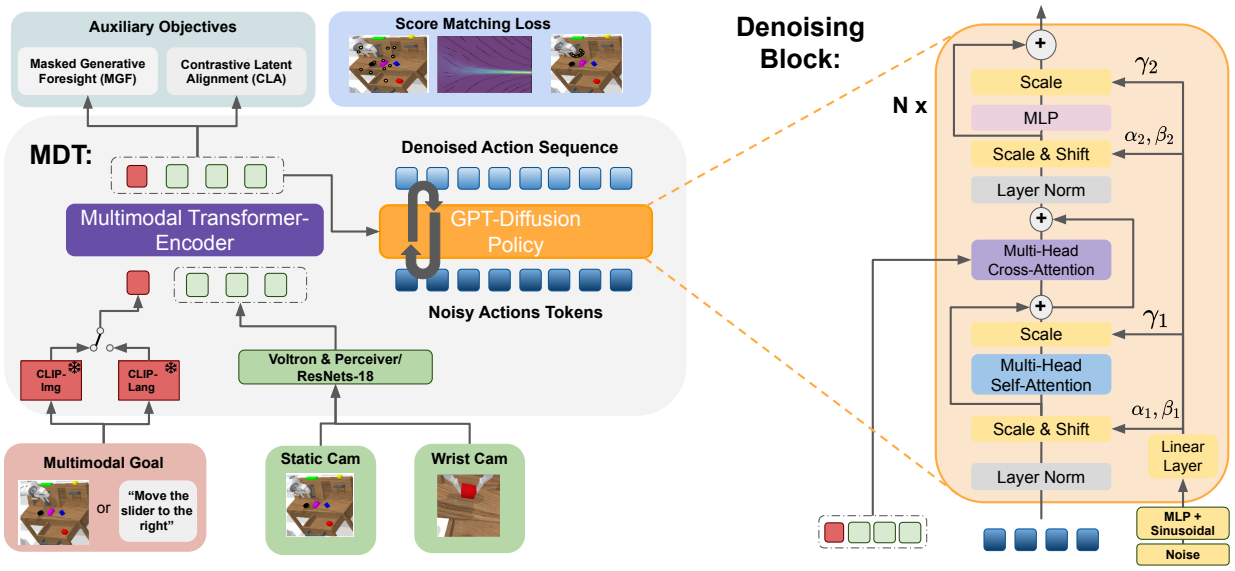

Fig. 3: (Left) Overview of the proposed multimodal Transformer-Encoder-Decoder Diffusion Policy used in MDT. (Right) Specialized Diffusion Transformer Block for the Denoising of the Action Sequence. MDT learns a goal-conditioned latent state representation from multiple image observations and multimodal goals. The camera images are either processed with frozen Voltron Encoders and a Perceiver or using 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 [22] 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.

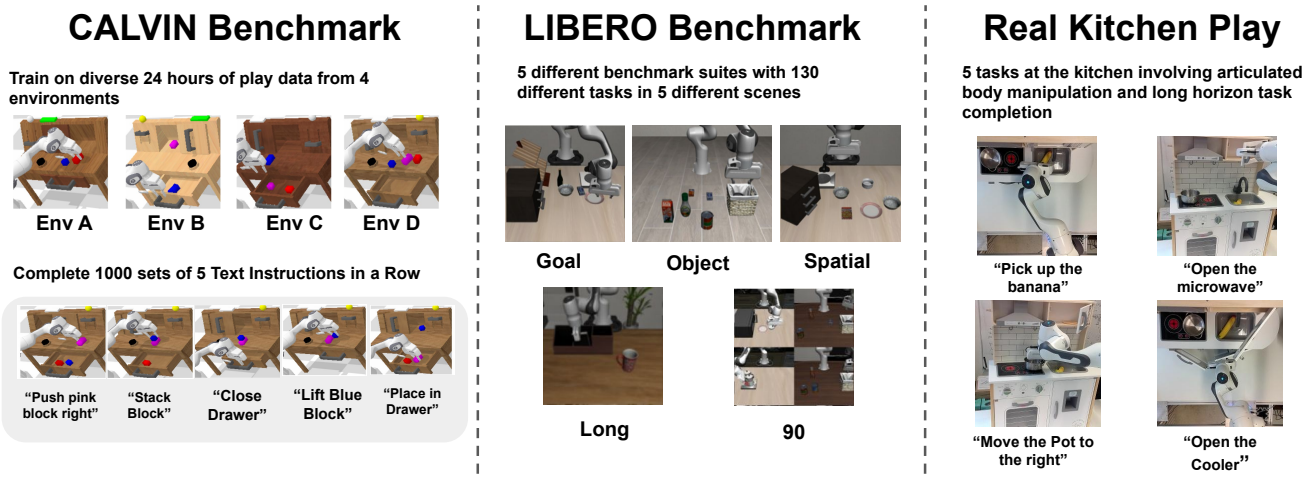

Fig. 4: Overview of the different environments used to test MDT: (Left) CALVIN Benchmark consisting of four environments each with unique positions and textures for slider, drawer, LED, and lightbulb. (Middle) Overview of the different tasks and scene diversity in the LIBERO benchmark, which is divided 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.

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	✓	✓	0.988	0.959	0.9115	0.861	0.794	4.51 ± 0.03

TABLE VI: 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. The results are reported over 1000 rollouts averaged over 3 seeds.

	CALVIN ABCD	LIBERO-Spatial
1	4.19 ± 0.03	64.4 ± 0.4
3	4.25 ± 0.04	67.5 ± 0.2
9	4.22 ± 0.06	65.6 ± 0.5

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

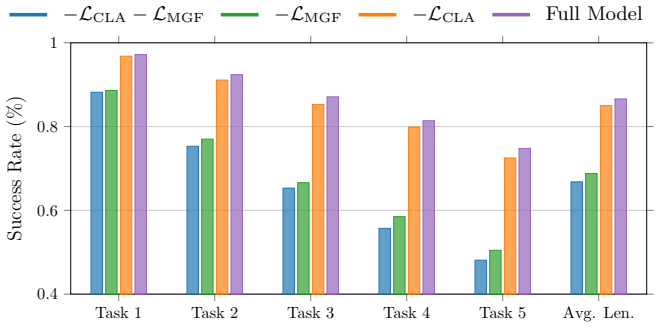

Fig. 5: Study on the performance of our proposed Masked Generative Foresight Loss and the Contrastive Latent Alignment Loss for our proposed MDT policy. We analyse the impact of both auxiliary tasks on the ABCD CALVIN challenge. The results show the average rollout length over 1000 instruction chains averaged over 3 seeds.

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).

G. Real Robot Experiments

For our real robot experiments, we collect approx. 2.5 hours of play data with 3 different persons, that were instructed to solve the tasks in a random order. We train all models with the collected data with a random geometric sampling of future frames to get goal images. Each policy was then trained for roughly 24 hours using a 24 hours on a small cluster with 4 GPUs. For evaluation purposes, we identified the most effective iteration of each model based on the lowest validation loss, except for the MT-ACTs models. For these, we selected the last epoch since our prior experience with the CALVIN model indicated an improvement in performance even when the validation loss began to rise again. To test all policies, we roll out each one ten times, using the same instruction chain with goal images or language annotations. Each chain consists of solving five tasks in either language or image goals. Example rollouts of these experiments are visualized in Figure 6.

We investigate research question (III), by assessing the ability of MDT to learn language-guided manipulation from partially labeled data in a real-world setting. MDT is evaluated on a real-world play kitchen setup with a 7 Degree-of-Freedom Franka Emika Panda Robot. The setup incorporates two static RGB cameras: one positioned above the kitchen for a bird's-eye view, and another placed on the left side of the robot. The experiments consist of five distinct tasks involving pick-and-place actions, door opening, and object manipulation. We annotate 20 sub-sequences of every task with several short language descriptions to create a partially annotated dataset. To enrich training diversity, some of these descriptions were generated by GPT-4. During training a single description per labeled sub-sequence is sampled. Our dataset encompasses around 2.5 hours of interactive play data. We evaluated both MT-ACT and MDT with an action sequence length of 20, which was optimal for performance on the physical robot. The models are tested with long-horizon rollouts, requiring the completion of five tasks in sequence. The models were provided 5 subgoals represented either as images or language descriptions. The experiments are detailed in Table X. The results highlight MDT's robust performance, particularly in learning from sparse labeling. MDT was successful in completing 4 tasks in sequence and showed proficiency in understanding goals expressed through language or images. The results further emphasise the importance of MGF to learn effectively from partially labeled datasets, since the MDT variant without this auxiliary objective showed a reduced performance.

Train→Test	Method	No. Instructions in a Row (1000 chains)					
		1	2	3	4	5	**Avg. Len.**
D→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%**	**83.2%**	**71.7%**	**60.5%**	**50.6%**	**3.60±(0.05)**

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

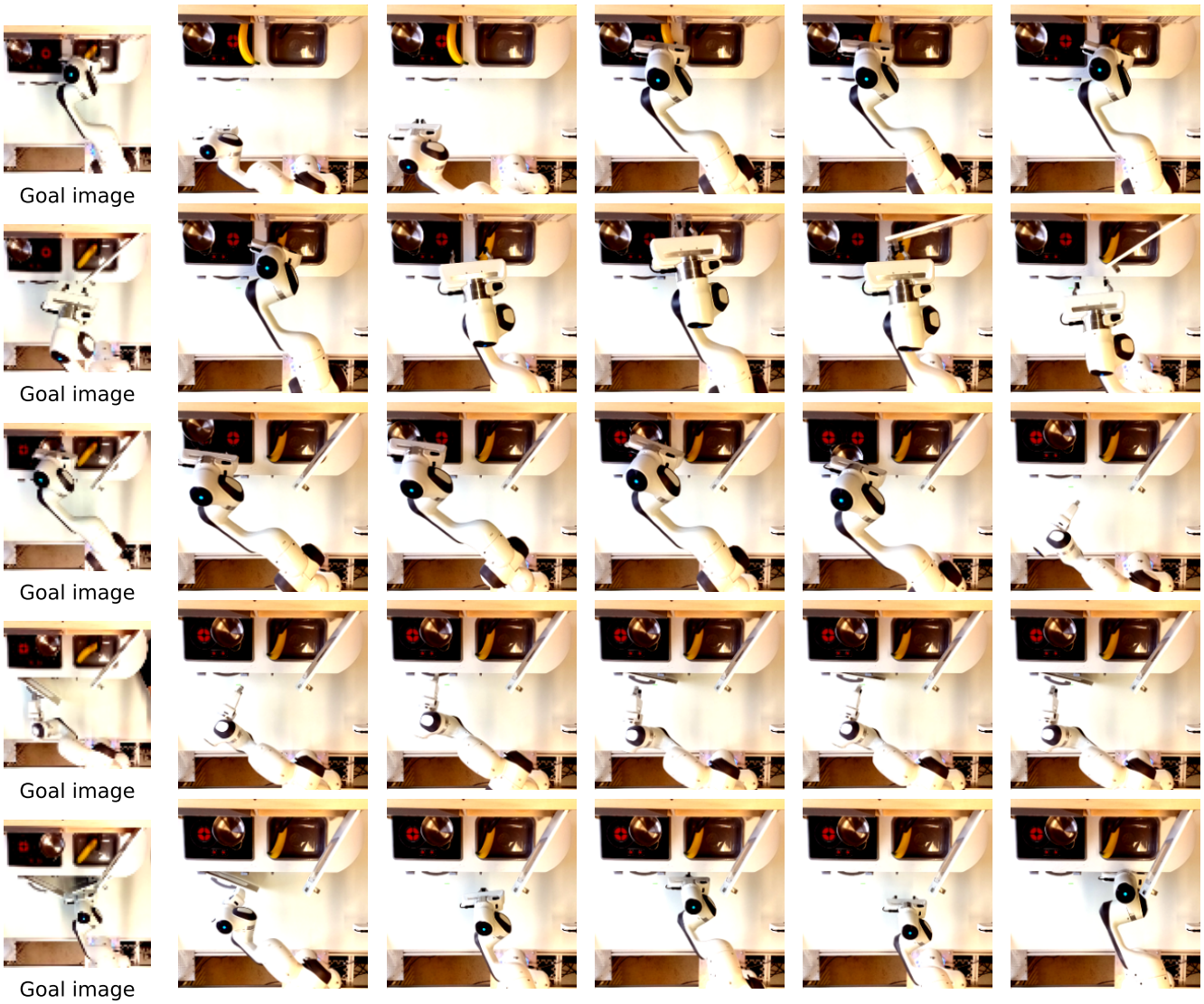

Fig. 6: Real Robot rollouts with goal image conditioning. The first column shows the goal image used for the rollout. 4 out of 5 tasks are successful. The robot fails to open the freezer door and accidentally closes the oven door.

Method	Language Annotation	$\mathcal{L}_{\text{CLA}}$	$\mathcal{L}_{\text{MGF}}$	Spatial	Object	Goal	Long	90	Average
Transformer-BC [68]	100 %	×	×	71.83 ± 3.7	71.00 ± 7.9	76.33 ± 1.3	24.17 ± 2.6	-	-
Distill-D	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	68.8 ± 2.2
	2%	✓	×	74.3 ± 0.8	87.5 ± 2.7	71.5 ± 3.5	63.9 ± 4.5	66.9 ± 1.7	**72.8 ± 2.6**
	2%	×	✓	67.5 ± 2.1	87.5 ± 2.6	69.3 ± 2.5	61.8 ± 2.5	62.6 ± 1.0	69.7 ± 2.1
	2%	✓	✓	**78.5 ± 1.5**	**87.5 ± 0.9**	**73.5 ± 2.0**	57.1 ± 0.8	**67.2 ± 1.1**	72.5 ± 1.5

TABLE IX: 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.

Goal Modality	Model	Task 1	Task 2	Task 3	Task 4	Task 5	Avg. Rollout Length
Language	MT-ACT	50%	50%	0%	0%	0%	1.00 ± (1.00)
	MDT	40%	40%	0%	0%	0%	0.80 ± (0.98)
	MDT + $\mathcal{L}_{\text{MGF}}$	90%	80%	10%	0%	0%	**1.80 ± (0.75)**
Images	MT-ACT	50%	40%	20%	10%	0%	1.20 ± (1.40)
	MDT	60%	30%	30%	10%	0%	1.30 ± (1.42)
	MDT + $\mathcal{L}_{\text{MGF}}$	100%	90%	90%	0%	0%	**2.80 ± (0.60)**

TABLE X: Average rollout length of different policies evaluated in our real robot play kitchen. We present the average performance of various policies tested on 10 long horizon instruction chains.

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.

H. Additional Ablation Studies

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 [27]. Thus, we use it as the default masking rate across all experiments in the paper. Further details of this analysis are provided in Table V 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 the second-best performance, it is also associated with increased variance in results. Further results of these investigations are presented in Figure VII in the Appendix.

Transformer Architecture. MDT is tested against two Diffusion Transformer architectures previously described in [12]. The ablations are visualized in Figure 2 of the Appendix. These comparisons are conducted on the CALVIN ABCD→D challenge, with detailed results featured in VI 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 [12], 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.

I. 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 reduce the performance on LIBERO-Long 2) The Contrastive Loss requires careful filtering of negative goal samples to use its full potential 3) Diffusion Policies require multiple forward passes to generate an action sequence which results in lower inference speed compared to non-diffusion approaches.