
EmbodiedMAE: A Unified 3D Multi-Modal Representation for Robot Manipulation

Zibin Dong[♡], Fei Ni[♡], Yifu Yuan[♡], Yinchuan Li[◇], Jianye Hao*^{♡, ◇}
[♡]Tianjin University, [◇]Huawei Noah’s Ark Lab

Abstract

We present EmbodiedMAE, a unified 3D multi-modal representation for robot manipulation. Current approaches suffer from significant domain gaps between training datasets and robot manipulation tasks, while also lacking model architectures that can effectively incorporate 3D information. To overcome these limitations, we enhance the DROID dataset with high-quality depth maps and point clouds, constructing DROID-3D as a valuable supplement for 3D embodied vision research. Then we develop EmbodiedMAE, a multi-modal masked autoencoder that simultaneously learns representations across RGB, depth, and point cloud modalities through stochastic masking and cross-modal fusion. Trained on DROID-3D, EmbodiedMAE consistently outperforms state-of-the-art vision foundation models (VFs) in both training efficiency and final performance across 70 simulation tasks and 20 real-world robot manipulation tasks on two robot platforms. The model exhibits strong scaling behavior with size and promotes effective policy learning from 3D inputs. Experimental results establish EmbodiedMAE as a reliable unified 3D multi-modal VFM for embodied AI systems, particularly in precise tabletop manipulation settings where spatial perception is critical.

1 Introduction

Pre-trained Vision Foundation Models (VFs) have made remarkable progress in visual understanding [6, 28, 15, 45, 26, 25, 1, 49], becoming essential components for embodied AI systems [27, 19, 3, 24, 44, 9, 22]. As research increasingly demonstrates that 3D spatial understanding can significantly improve robot manipulation capabilities [44, 17, 21, 46], the demand for effective 3D VFs has grown substantially. 3D information provides critical spatial context, enabling robots to accurately localize targets, avoid collisions, and execute complex manipulations. However, despite this increasing need, existing models fall short of meeting requirements.

There are two primary reasons behind the lack of suitable 3D VFs for embodied AI. *First, a significant domain gap exists in training data.* Mainstream 3D VFs are trained on outdoor or indoor static scenario datasets [16, 47, 31, 41, 42]. These models operate at spatial scales incompatible with tabletop manipulation, where precise understanding within a 20 cm to 1.5 m range is crucial and results in a weak understanding of robot-object interactions [44]. While training 3D embodied-specific VFs from scratch on robot manipulation datasets seems promising, these efforts are hampered by extremely limited training data [49, 32]. For example, OpenX Embodiment [36], despite being the largest embodied manipulation dataset, contains minimal high-quality 3D information, making it insufficient for effective pre-training. *Second, there is a lack of efficient and scalable model architectures for 3D perception.* Simply integrating 3D information without careful design often degrades robot operation capabilities rather than enhancing them. For example, many advanced 3D VFM architectures demonstrate unexpectedly poor performance in robot manipulation settings, sometimes even underperforming simple MLPs [44, 48].

*Corresponding author: Jianye Hao (jianye.hao@tju.edu.cn).

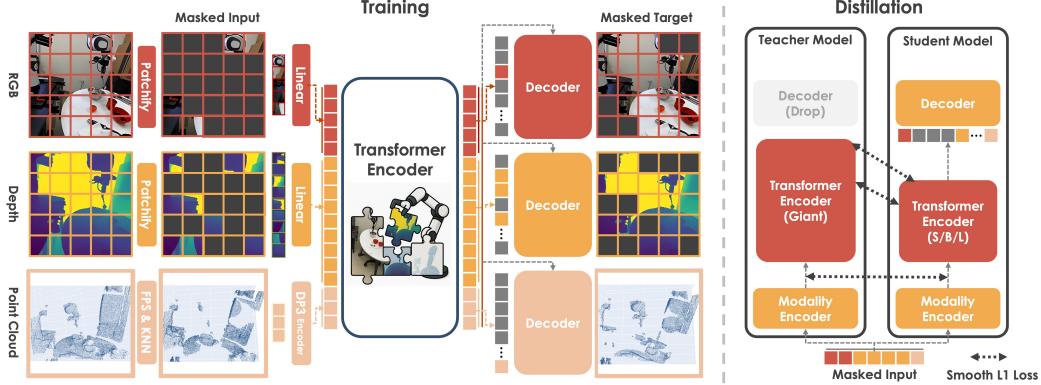


Figure 1: Overview of EmbodiedMAE Pre-training. We pre-train a ViT-Giant scale multi-modal MAE on the large-scale DROID-3D robot manipulation dataset. We fix the total number of unmasked patches across RGB, depth, and point cloud modalities. The mask ratio allocated to each modality is stochastically sampled. After the Giant model pre-training, we distill it to obtain our Small/Base/Large scale models.

To address these challenges, we propose EmbodiedMAE, a unified 3D multi-modal representation learning framework specifically designed for embodied AI. We first enhance the original DROID dataset [18] by extracting high-quality metric depth maps and point clouds for each frame using ZED SDK temporal fusion and AI-augmented enhancement. This creates DROID-3D, a large-scale 3D robot manipulation dataset containing 76K trajectories (350 hours) of high-fidelity interaction data. This dataset provides the scale and quality needed for effective pre-training while maintaining domain compatibility with manipulation tasks. We then develop a multi-modal masked autoencoder that simultaneously learns representations across RGB images, depth maps, and point clouds through stochastic masking and cross-modal fusion. By masking different proportions of each modality and using explicit modal fusion in the decoder, our model learns to infer across modalities, developing powerful spatial perception capabilities and object-level semantic understanding (Figure 3).

To thoroughly validate our representation model, we conduct extensive evaluations across diverse settings: 40 and 30 simulation tasks from the LIBERO benchmark [23] and the MetaWorld benchmark [43], 10 real-world tasks on the low-cost open-source SO100 robot platform [5], and 10 tasks on the high-performance xArm robot platform. We use a scaled-down RDT [24] model as the policy backbone to simulate the performance of VFM in advanced VLA training, and compare EmbodiedMAE against various categories of state-of-the-art (SOTA) VFM, including vision-centric models, language-augmented models, embodied-specific models, and 3D-aware models. Our experiments demonstrate that EmbodiedMAE consistently outperforms all baseline VFM in both training efficiency and final performance, exhibits strong scaling behavior with model size, and effectively promotes policy learning from 3D input. These findings establish EmbodiedMAE as a reliable foundation model for embodied AI applications requiring robust 3D visual understanding.

Our contributions can be summarized as follows:

- We present EmbodiedMAE, a unified 3D multi-modal representation learning framework for embodied AI that effectively integrates RGB, depth, and point cloud modalities. It achieves SOTA performance in both RGB-only and multi-modal settings while maintaining computational efficiency and scaling properties.
- We introduce DROID-3D, a high-quality, large-scale 3D robot manipulation DROID supplement dataset containing 76K trajectories (350 hours) of interaction data with synchronized RGB, depth maps, and point clouds. Unlike previous works that processed only subsets or used low-quality AI-estimated depth, we provide temporally consistent depth by ZED SDK processing, creating a valuable resource for 3D robot learning research.
- We establish comprehensive evaluation benchmarks for embodied representation learning across diverse settings: simulation tasks from LIBERO and MetaWorld, real-world tasks on a low-cost open-source robot (SO100), and tasks on a high-performance robot (xArm). Our results demonstrate consistent performance improvements across these varied platforms, validating the model’s generalization capabilities.

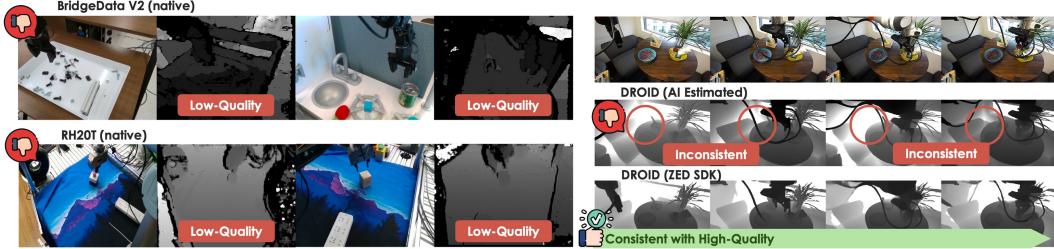


Figure 2: **Depth Quality Comparison.** We evaluate depth data quality across several mainstream large-scale embodied AI datasets. Both BridgeDataV2 and RH20T exhibit unreliable and noisy depth information. While prior work has explored the use of AI models for depth estimation, we observe that such methods lack temporal consistency. In contrast, our solution, ZED SDK processing, achieves superior and consistent depth quality.

2 Methodology

2.1 3D Data Collection

Effective pre-training of our model necessitates a large-scale 3D robot manipulation dataset. We conduct a systematic evaluation of depth data quality across several mainstream large-scale embodied AI datasets, primarily including BridgeDataV2 [37], RH20T [12], and DROID [18], as illustrated in Figure 2. We find significant limitations in existing datasets: BridgeDataV2 contains only 13% data with 3D information, with available depth maps being of insufficient quality; RH20T exhibits similar issues with unreliable and noisy depth data; while DROID includes stereo image recordings but lacks readily usable 3D annotations. Several previous approaches attempted to address this by estimating depth from 2D images using AI models. For instance, SPA [49] employs CrocoV2-Stereo [39] to estimate depth for approximately 1/15 of the DROID dataset. We observe that such methods lack precision and temporal consistency, making them unable to accurately capture fine-grained details during robot-object interactions, which are essential for manipulation tasks.

To overcome these challenges, we leverage the fact that the raw DROID dataset preserves ZED camera recordings, which can be processed using ZED SDK to extract high-quality depth information. The ZED SDK integrates multiple techniques that significantly improve depth quality, including temporal fusion to reduce noise and increase consistency, AI-augmented enhancement to refine stereo matching in textureless regions, and hardware-calibrated metric depth to provide accurate absolute distance measurements. Using these high-quality depth maps, we further extract point clouds with the camera’s intrinsic matrix. We apply farthest point sampling (FPS) to downsample them to 8,192 points, striking a balance between computational efficiency and geometric fidelity. Unlike SPA’s approach of processing only a subset of the DROID dataset, we process the complete collection of 76K trajectories (350 hours of interaction data), requiring nearly 500 hours of processing time. Due to these significant improvements in data quality and coverage, we construct and release DROID-3D as a supplementary resource to the original DROID dataset. We believe it will serve as a valuable resource for pre-training 3D VLA models and foster innovative research in embodied AI, particularly for applications requiring precise spatial understanding for manipulation tasks.

2.2 Multi-Modal Encoder

EmbodiedMAE processes three modalities commonly used in robot perception: RGB images, depth maps, and point clouds. Given the robot observation of RGB image $I \in \mathbb{R}^{3 \times H \times W}$, depth $D \in \mathbb{R}^{1 \times H \times W}$, and point cloud $P \in \mathbb{R}^{M \times 3}$, we first use modal-specific patchifiers to project them into patches $\bar{I}, \bar{D}, \bar{P} \in \mathbb{R}^{L \times C}$. Then we draw a random binary mask for each modality $m_I, m_D, m_P \in \{0, 1\}^L$, and obtain two complementary masked views $I_1 = \bar{I}[m_I], I_2 = \bar{I}[1 - m_I]$, similar for D and P . We use a Vision Transformer (ViT) f to process the unmasked patches and obtain the joint representation $h = f(I_1, D_1, P_1)$.

Masking Strategies. Effective masked autoencoding requires masking a large portion of input tokens during training, and the specific masking strategy has a significant impact on learned representations [1, 15]. Following Bachmann et al. [1], we fix the total number of unmasked patches across all modalities, i.e., the number of ones in (m_I, m_D, m_P) is fixed, and allocate them according to a symmetric Dirichlet distribution: $(\lambda_I, \lambda_D, \lambda_P) \sim \text{Dir}(\alpha)$, where $\lambda_I + \lambda_D + \lambda_P = 1$ and each

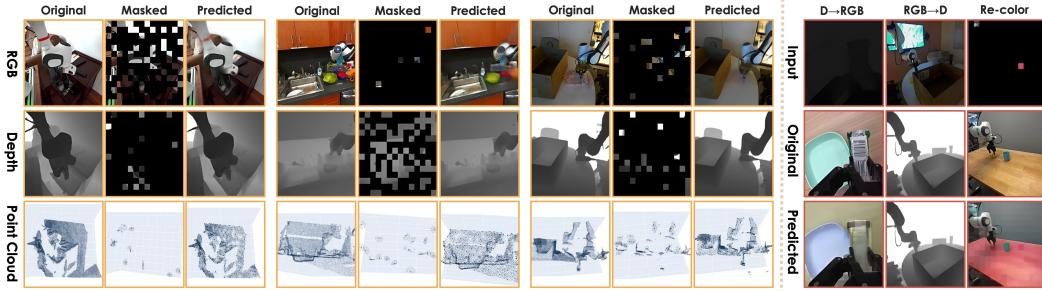


Figure 3: **EmbodiedMAE Visual Predictions.** We evaluate its visual predictions under three settings: **(a)** Two modalities are almost masked, leaving one modality as the major infer source (column 1-9). **(b)** Model predicts one modality from another one (column 10-11). **(c)** Model is allowed to see a modified RGB patch during depth-to-RGB prediction, where the color of the visible patch is altered (column 12).

$\lambda \geq 0$. The concentration parameter α controls the diversity of masking proportions. When $\alpha = 1$, the distribution is uniform over the simplex, assigning equal likelihood to all valid combinations. Lower values ($\alpha \ll 1$) tend to concentrate sampling on a single modality, while higher values ($\alpha \gg 1$) produce more balanced allocations across modalities. We intentionally avoid introducing any modality bias by keeping the distribution symmetric, aiming to maintain flexibility for a variety of downstream tasks and input configurations.

Modal Patchifiers. For RGB and depth maps, we break them into 16×16 -size patches, i.e., $L = \frac{H \cdot W}{16^2}$, and we incorporate 2D sine-cosine positional embeddings after a linear projection [11, 35]. For point clouds, we apply Farthest Point Sampling (FPS) to select N cluster centers, and then use K-Nearest Neighbors (KNN) to group each center with its K nearest neighbors, forming N point groups of $K + 1$ points each, i.e., $L = N$. Each group is normalized and encoded using a DP3 encoder [44] to generate token embeddings, while each group center is processed by an MLP to create positional embeddings [29]. We omit explicit modality-type embeddings, as the bias term in each projection layer implicitly encodes modality-specific information. These tokens are masked, concatenated, and passed to the ViT encoder to produce the joint representations.

Transformer Encoder. We implement the same ViT structure as DINOv2 [28], with the exception of removing the [CLS] token. This design choice allows us to initialize the ViT directly from DINOv2 pre-trained weights, thereby enhancing its general capabilities.

2.3 Multi-Modal Decoder

The decoder is only used during EmbodiedMAE training, where it reconstructs the masked portions of each modality based on the visible tokens and learned [MASK] tokens.

Specifically, the decoder employs cross-attention to enable explicit fusion across modalities. Visible tokens from each modality are projected and concatenated with [MASK] tokens, then augmented with positional embeddings to form the query sequence. Meanwhile, all visible patches are projected and enhanced with modality encodings to form the key and value sequences. The fused features are then fed into a smaller, modality-shared ViT decoder to produce the final hidden states. Modality-specific MLP heads generate the reconstruction outputs: masked RGB and depth patches, and normalized point coordinates for point cloud groups. Suppose that $(h_I, h_D, h_P) = f(I_1, D_1, P_1)$ are modality representations, the decoder outputs can be expressed as $g_I(h_I, h)$, $g_D(h_D, h)$, and $g_P(h_P, h)$, corresponding to each modality. Notably, our design shares transformer components across modalities, reducing computational cost by approximately a factor of three. We adopt a simple mean square error (MSE) loss:

$$\mathcal{L}_{\text{MAE}} = \mathbb{E}_{(I, D, P) \sim \mathcal{D}, \text{Dir}(\alpha)} \left[\underbrace{\|g(h_I, h) - I_2\|^2}_{\text{RGB}} + \underbrace{\|g(h_D, h) - D_2\|^2}_{\text{Depth}} + \underbrace{\|g(h_P, h) - P_2\|^2}_{\text{PointCloud}} \right], \quad (1)$$

where the decoder outputs $g_I(h_I, h)$ and $g_D(h_D, h)$ are l_2 -normalized, while $g_P(h_P, h)$ is group center-normalized, following MAE’s finding that normalized targets yield better performance. [15].

2.4 Model Distillation

Following Oquab et al. [28], we first train a ViT-Giant EmbodiedMAE model from scratch on the DROID-3D dataset, then distill it into Small, Base, and Large variants. Both teacher and student models receive identical masked inputs (I_1, D_1, P_1), with the teacher model kept entirely frozen. Rather than simply copy the final outputs, we apply feature-level supervision at strategically selected network depths to ensure comprehensive knowledge transfer. Specifically, we align features at three critical positions in the network hierarchy: (Bottom) immediately after the modal patchifiers to capture low-level perceptual features, (Top) at the final hidden layer to preserve high-level semantic understanding, and (Middle) at a middle layer positioned at 3/4 of the encoder depth to transfer intermediate representations [2] (For example, when distilling from a 24-layer ViT-L teacher to a 12-layer ViT-B student, the 9th layer of the student aligns with the 18th layer of the teacher.). We adopt trainable linear projections before computing alignment losses to accommodate dimensional differences between teacher and student features. Formally, we denote the feature alignment pairs $(y^j, h^j) \in A$, where y^j and h^j represent the j -th pair of hidden states from teacher and student models, respectively, and l^j is the linear projector. The feature alignment loss can be expressed as:

$$\mathcal{L}_{\text{Align}} = \sum_{(y^j, h^j) \in A} \text{SmoothL1}(y^j, l^j(h^j)). \quad (2)$$

We train student models by jointly optimizing the standard multi-modal MAE reconstruction loss and the feature alignment loss (Figure 1, Distillation part):

$$\mathcal{L}_{\text{Distill}} = \mathcal{L}_{\text{MAE}} + \beta \cdot \mathcal{L}_{\text{Align}}, \quad (3)$$

where $\beta > 0$ controls the balance between mask autoencoding and feature alignment. This approach enables our smaller models to achieve performance closer to the Giant model while maintaining computational efficiency, making them practical in resource-constrained robotics applications.

2.5 Put All Together

Building on our architectural design described above, we first pre-train the Giant-scale model and subsequently distill it into more computationally efficient Small, Base, and Large variants on the DROID-3D dataset. We employ AdamW optimizer with a weight decay of 0.01. The base learning rate is set at 1.5e-4, incorporating an initial warmup period followed by a cosine schedule decay. We apply a 0.1 gradient norm clip to stabilize training. All computational workflows utilize bfloat16 precision, which substantially reduces memory requirements and computational costs while maintaining numerical stability. During the pre-training phase, we maintain 96 unmasked patches across all modalities, representing approximately 1/6 of the total patch count. For the distillation phase, we further reduce the number of unmasked patches to 60, approximately 1/10 of the total. This extremely aggressive masking approach significantly decreases training costs without compromising representational quality, as the student models benefit from the teacher’s already robust understanding of multi-modal relationships. The efficient training strategy enables us to complete Giant model pre-training on 8 NVIDIA L40 48G GPUs and the distillation phases on 4 NVIDIA GeForce RTX 4090 24G GPUs. We provide hyperparameter configuration list in Table 6.

Our codebase follows the Huggingface Transformers [40] convention, making EmbodiedMAE highly user-friendly. It ensures that researchers can easily incorporate our models into existing robotics pipelines with minimal adaptation effort. A simple usage example is illustrated in Figure 4.

3 Experiments

In this section, we present evaluation results of EmbodiedMAE across both simulation and real-world robotic manipulation tasks. Our experiments are designed to address three key research questions:

(RQ1) Does EmbodiedMAE learn features that integrate information across different modalities?

```
from embodied_mae import EmbodiedMAEModel
model = EmbodiedMAEModel.from_pretrained("/path/to/ckpt")
rgb_feature = model(rgb, None, None).last_hidden_states
# (b, 196, dim)
rgbd_feature = model(rgb, depth, None).last_hidden_states
# (b, 392, dim)
pc_feature = model(None, None, pc).last_hidden_states
# (b, 196, dim)
```

Figure 4: **Usage Example.** We follow the Huggingface Transformers convention to make EmbodiedMAE highly user-friendly and easy to integrate.

(RQ2) How does EmbodiedMAE perform compared to SOTA VFM in robot manipulation tasks?

(RQ3) Can EmbodiedMAE enable efficient robot learning in real-world environments for both low-cost and high-performance robot platforms?

3.1 Experimental Setup

Policy Network. To evaluate how effectively different VFM support advanced VLA models, we adopt a compact RDT [24] (approximately 40M parameters) as our policy network. This architecture has demonstrated excellent scalability and strong performance in diffusion-based policy learning. As shown in Figure 5, all baselines and EmbodiedMAE share the same architecture, ensuring fair comparison by isolating the visual representation component. See Appendix A.1 for more policy network details.

Baselines. To enable a comprehensive comparison, we benchmark against several SOTA VFM with diverse design principles: DINOv2-Large [28] (vision-centric), SigLIP-Large [45] (language-contrastive), R3M-Resnet50 [26], VC-1 [25], and SPA [49] (embodied-specific). Notably, SPA incorporates implicit 3D spatial priors during training, making it particularly relevant for comparison with our multi-modal approach.

Benchmarks. Our simulation evaluations are based on the LIBERO and MetaWorld benchmarks. LIBERO includes 40 tasks in four task suites: *Goal*, *Spatial*, *Object*, and *Long*. MetaWorld includes 30 tasks from various difficulty levels. For real-world experiments, we deploy the models on two robot platforms: The SO100 robot (low-cost, open-sourced, equipped with dual RGB cameras) evaluated on 10 tasks in suites: *Pick&Place*, *MoveTo*, *Wipe*, and *Unfold*; The xArm robot (higher-precision, equipped with one Intel RealSense L515 LiDAR camera) evaluated on 10 tasks in suites: *Pick&Place*, *Pot*, *Pour*, and *Moka*. We show detailed task configurations in Appendix A.2.

3.2 MAE Predictions (RQ1)

To assess the ability of EmbodiedMAE to integrate information across modalities, we design a series of controlled experiments probing its cross-modal fusion capabilities. Our evaluation focuses on three settings: **(a) Extreme modality inference:** We mask most patches from two modalities, leaving primarily one modality as the inference source (Figure 3, columns 1-9). **(b) Cross-modal translation:** We test the model’s ability to predict one entire modality from another, specifically RGB from depth (column 10) and depth from RGB (column 11). **(c) Re-coloring:** We allow the model to see a deliberately altered RGB patch during depth-to-RGB prediction (column 12), where the color of the visible patch is modified to assess semantic understanding. Our results demonstrate that EmbodiedMAE effectively leverages available modalities to reconstruct missing information, suggesting strong cross-modal alignment. In column 10, the predicted RGB from depth lacks precise color information but maintains structural fidelity, indicating the model has learned to separate geometric and appearance features. Similarly, in column 11, depth predictions from RGB show smoothed object boundaries compared to ground truth, revealing a learned prior for depth continuity. Most notably, in the re-coloring setting (column 12), when injecting an altered RGB patch during depth-to-RGB reconstruction, only the corresponding object (table) adopts the modified color while surrounding elements (background, robot, cup) maintain their original appearance. This suggests EmbodiedMAE has implicitly learned object-level semantic segmentation and can propagate semantic information based on contextual cues, despite never being explicitly trained for segmentation.

These visualizations collectively demonstrate that EmbodiedMAE possesses strong multi-modal fusion capabilities, enabling it to enhance spatial understanding in 3D embodied perception tasks.

3.3 Overall Comparison (RQ2)

In this section, we evaluate SOTA VFM baselines, EmbodiedMAE, and several its variants (in terms of model scale and input modality) on the LIBERO and MetaWorld benchmark. We report learning

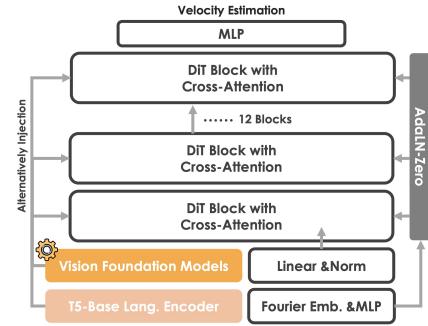


Figure 5: **Policy Network for All VFM.** We adopt a compact RDT as the policy network, in which only VFM are modular.

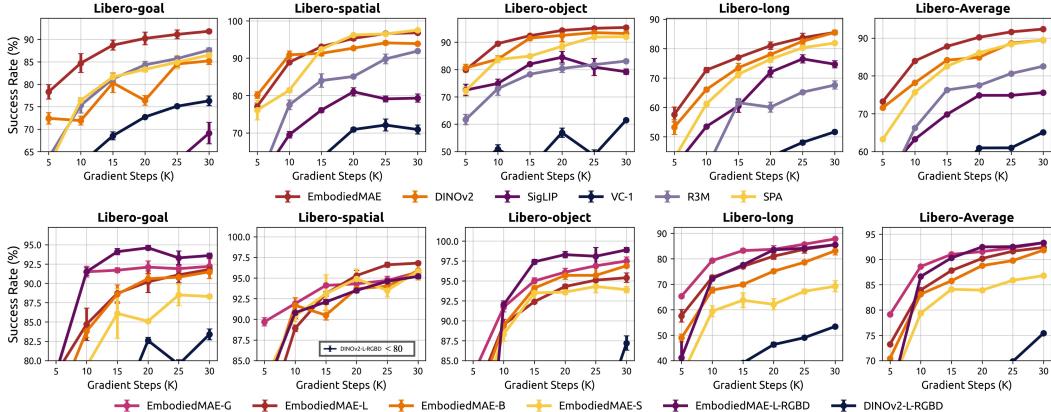


Figure 6: **Learning curve on LIBERO benchmark.** Each task is evaluated across 150 trials. Our model surpasses all baselines on the LIBERO benchmark and demonstrates scaling capabilities, with performance increasing proportionally with model size. Our model effectively leverages 3D information to further enhance policy performance, whereas naively incorporating depth information results in performance degradation.

Table 1: **Success rate on MetaWorld benchmark.** We report the average success rate for each difficulty level. The numerical suffix following each level indicates the number of tested tasks. Note that **Average** row represents the average across all tasks rather than the three difficulty levels. Highest scores are emphasized with bold.

MetaWorld Difficulty Level	R3M	SigLIP	DINOv2	SPA	EmbodiedMAE	DINOv2 -RGBD	EmbodiedMAE -RGBD
Easy (18)	74.1	76.4	79.8	80.9	<u>81.8</u>	61.9	85.2
Medium (9)	28.1	32.7	57.1	<u>62.8</u>	60.4	35.6	63.2
Very Hard (3)	49.8	14.0	56.4	<u>55.8</u>	57.8	65.6	<u>61.6</u>
Average	57.9	57.0	70.7	<u>73.0</u>	<u>73.0</u>	54.4	76.2

curves on LIBERO in Figure 6 and success rate on MetaWorld in Section 3.3. Unless otherwise specified, “EmbodiedMAE” refers to the Large-scale, RGB-only variant.

Finding 1: EmbodiedMAE consistently outperforms all baseline VFM in terms of both training efficiency and final performance. Among the baselines, SPA and DINOv2 are the most competitive ones. SPA shows score gains on tasks where spatial understanding is crucial, e.g., LIBERO-Spatial and MetaWorld, and performs comparably to DINOv2. The language-contrastive model, SigLIP, performs poorly across all embodied tasks, consistent with findings from Zhu et al. [49]. R3M and VC-1, although specifically designed for robot learning, do not demonstrate clear advantages.

Finding 2: EmbodiedMAE exhibits strong scaling behavior with model size. Performance improves monotonically as model capacity increases. Among all the variants, only the Small variant shows unstable performance on LIBERO-Goal and LIBERO-Object suites. The Base and Large models achieve similar performances, with the Large model slightly ahead. The Giant model consistently delivers superior performance, particularly in training efficiency. These results suggest EmbodiedMAE to be an effective training paradigm for scaling multi-modal representation learning.

Finding 3: EmbodiedMAE promotes policy learning from 3D input. When provided with RGBD inputs, EmbodiedMAE establishes a substantial performance gap over other baselines on both LIBERO and MetaWorld benchmarks. Remarkably, our Large-scale RGBD model even outperforms the Giant-scale RGB-only model on LIBERO-Goal and LIBERO-Object suites, and performs comparably on average across the LIBERO benchmark. In contrast, adding a trainable depth branch for DINOv2 (See Appendix A.3 for details of this variant) can degrade performance relative to RGB-only input, consistent with observations in Zhu et al. [48]. These findings establish EmbodiedMAE as a reliable VFM for scenarios requiring 3D visual understanding.

3.4 Real-World Experiments (RQ3)

To further assess generalization in practical settings, we conduct real-world evaluations on two robot platforms: the low-cost, open-source SO100 [5] and the high-performance xArm. We show quantitative results in Figure 8, and rollout visualizations in Figure 7.

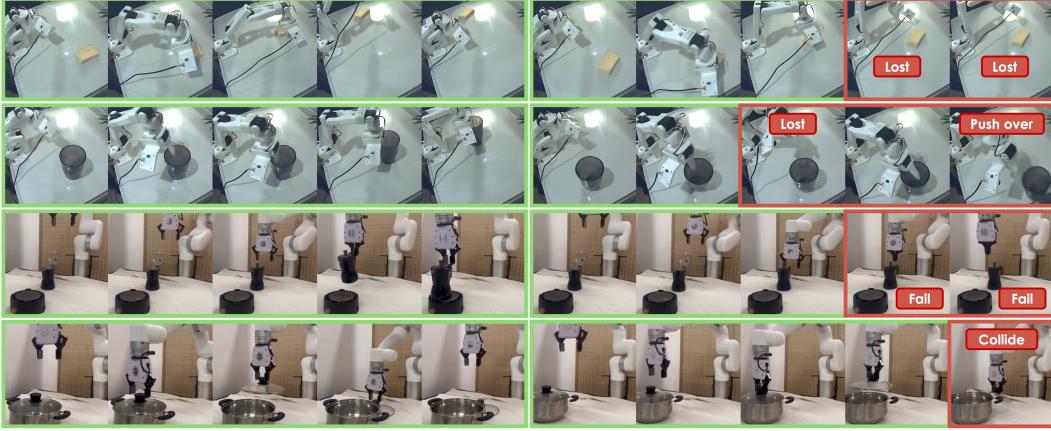


Figure 7: **Successful rollouts of EmbodiedMAE (Left) and typical failure cases of baselines (Right).** Baseline models often fail due to inaccurate localization, leading to object loss, grasp failure, or collisions. In contrast, EmbodiedMAE benefits from stronger spatial perception and avoids such errors more effectively.

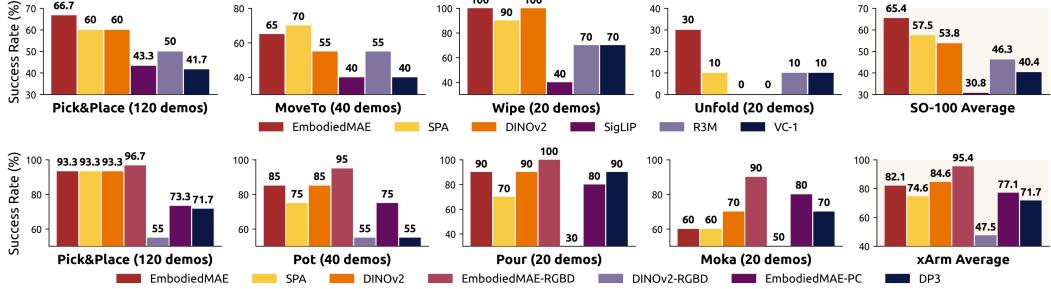


Figure 8: **Evaluation results on SO100 and xArm platforms.** Each task is evaluated across 10 trials. On the SO100 platform, our model outperformed all baselines in the RGB-only setting. On the xArm platform, our model achieved comparable performance to SOTA baselines in the RGB-only setting, while significantly surpassing baselines in both RGBD and Point Cloud settings.

Finding 1: EmbodiedMAE maintains SOTA performance in real-world robot manipulation. EmbodiedMAE consistently achieves SOTA performance across real-world manipulation tasks, particularly those requiring strong spatial understanding. With multi-modal inputs, EmbodiedMAE further improves policy learning performance: EmbodiedMAE-RGBD and EmbodiedMAE-PC both surpass naïve fusion baselines such as DINOv2-RGBD (Appendix A.3) and DP3 [44], highlighting the effectiveness of our design in promoting robust 3D perception for real-world robotics applications.

Finding 2: 3D information plays a critical role in robot manipulation. Incorporating 3D inputs significantly improves task success rates. We observe that most failures in baseline models stem from localization errors, causing object loss, grasp failures, or collisions. EmbodiedMAE-RGBD, benefiting from enhanced spatial understanding, avoids these issues more reliably (see Figure 7). The choice of 3D modality also matters. Although prior works [21, 44, 48] have highlighted the compactness and training efficiency of point cloud (PC) representations, we find their practical effectiveness is hindered by sensor noise from object reflectivity and lighting variations. Consequently, in our experiments, PC-based policies even underperform RGB-only inputs. In contrast, the RGBD setting, where depth serves as an auxiliary cue, yields better performance and is more robust to depth noise. This suggests that effective post-processing of point clouds is essential for leveraging them reliably; otherwise, RGBD inputs offer a more dependable alternative.

3.5 Ablation Studies

Due to the prohibitive cost of ViT-Giant pre-training, our ablation studies focus on model distillation insights. We evaluate masking ratio, feature alignment, and loss ratio on the LIBERO benchmark, reporting average success rates in Table 2, with default settings underlined. **(1) Masking Ratio:** Our default configuration sets 60 unmasked patches, approximately masking ratio of 90%. We test

70%, 80%, and 100% ratios (100% representing training with only feature alignment loss). Results indicate performance insensitivity to masking ratio, though ratios <100% perform better, suggesting feature alignment’s predominant role while mask autoencoding provides additional benefits. **(2) Feature Alignment:** By default, we implement feature alignment at three positions (see Section 2.4). Sequential removal of alignment points reveals diminishing impact from Top to Bottom, with each component contributing positively to model performance. **(3) Loss Ratio:** With default $\beta = 1$, we test $\beta = 0.5/2.0/4.0$. Results show performance robustness across β values, with slight degradation at $\beta < 1.0$, confirming feature alignment necessity, consistent with findings in [2].

4 Related Works

Vision Foundation Models are models trained on large-scale data in a self-supervised or semi-supervised manner that can be adapted for several other downstream tasks [4]. Beyond conventional image classification, these models have shown strong transfer capabilities to tasks such as depth estimation [41, 42, 39], semantic segmentation, and robot control [27, 19, 24, 20]. Common pre-training techniques include contrastive learning [14, 7, 8], masked autoencoding [2, 34, 38, 13, 15], self-distillation [28, 6], and CLIP-style language-image contrastive learning [45, 33]. VFM greatly improve AI systems’ visual understanding.

Visual Representations for Embodied AI are crucial for enabling agents to perceive and interact with the physical world. Embodied perception must model robot-object interactions in dynamic environments, which general-purpose VFM trained on static images often lack. Several recent methods have attempted to bridge this gap by training models directly on robot datasets. However, the limited scale and quality of embodied data hinder their generalization. These embodied-specific models often fail to generalize as well as VFM trained on diverse in-the-wild datasets. As a result, many VLA models still rely on general-purpose VFM like DINOv2 [28, 19, 20] and SigLIP [45, 24, 19] for better generalization, prompting the need for dedicated large-scale embodied VFM pretraining.

3D Robot Learning has proven effective in improving both embodied agents’ training efficiency and manipulation success rate [44, 21, 48]. Properly introducing 3D visual inputs, such as depth or point clouds, often leads to better spatial understanding compared to RGB-only inputs. However, naively incorporating 3D information, e.g., adding an extra depth channel, may severely degenerate the model’s performance. Scalable native 3D multi-modal models remain largely absent in the current research landscape. EmbodiedMAE aims to address this gap by pre-training VFM on large-scale, embodied-specific datasets to facilitate the development of scalable and effective 3D VLA models.

5 Conclusion, Limitations, and Future Works

In this work, we introduce EmbodiedMAE, a unified 3D multi-modal representation learning framework designed for robot learning. We first construct DROID-3D, a high-quality, large-scale 3D robot manipulation DROID supplement dataset, containing 76K trajectories. Then we propose a multi-modal masked autoencoder architecture that fuses RGB, depth, and point cloud inputs through stochastic masking and cross-modal decoding. Trained on DROID-3D, our model, EmbodiedMAE, demonstrates superior spatial understanding, strong multi-modal fusion ability, and effective scaling behavior. It outperforms strong VFM baselines across 70 simulation tasks and 20 real-world tasks on two robot platforms (SO100 and xArm). We believe both the DROID-3D dataset and EmbodiedMAE provide a valuable resource for 3D robot learning research. Despite the strong performance, EmbodiedMAE remains solely a vision backbone and does not natively support language instruction as input. A promising future direction is to fully leverage the language and action annotations available in the DROID-3D dataset to train a vision-language backbone, or even develop a multi-modal VLA model for instruction-following general embodied agents.

Masking Ratio	0.7	0.8	<u>0.9</u>	1.0
	92.2	91.2	92.4	90.1
Feature Alignment	w/o Bottom	w/o Middle	w/o Top	All
	91.4	88.5	74.4	92.4
Loss Ratio β	0.5	<u>1</u>	2	4
	90.8	92.4	91.1	92.2

Table 2: **Ablation study on LIBERO.** We conduct ablation experiments on masking ratio, feature alignment, and loss ratio on the LIBERO benchmark and report the average success rate.

References

- [1] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision, ECCV*, 2022.
- [2] Yutong Bai, Zeyu Wang, Junfei Xiao, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. Masked autoencoders enable efficient knowledge distillers. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2023.
- [3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. In *arXiv preprint arXiv:2108.07258*, 2022.
- [5] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. <https://github.com/huggingface/lerobot>, 2024.
- [6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision, ICCV*, 2021.
- [7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. In *arXiv preprint arXiv:2003.04297*, 2020.
- [8] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. In *arXiv preprint arXiv:2104.02057*, 2021.
- [9] 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.
- [10] Zibin Dong, Yifu Yuan, Jianye HAO, Fei Ni, Yi Ma, Pengyi Li, and YAN ZHENG. Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, NIPS*, 2024.

- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations, ICLR*, 2021.
- [12] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning, RSS*, 2023.
- [13] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *Advances in Neural Information Processing Systems, NIPS*, 2022.
- [14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *arXiv preprint arXiv:1911.05722*, 2019.
- [15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.
- [16] Di Huang, Sida Peng, Tong He, Honghui Yang, Xiaowei Zhou, and Wanli Ouyang. Ponder: Point cloud pre-training via neural rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, 2023.
- [17] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *8th Annual Conference on Robot Learning, CoRL*, 2024.
- [18] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Young-woon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Khuong Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS 2024 Workshop: Data Generation for Robotics, RSS*, 2024.
- [19] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Open-VLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning, CoRL*, 2024.
- [20] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *arXiv preprint arXiv:2502.19645*, 2025.
- [21] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. In *arXiv preprint arXiv:2503.07511*, 2025.
- [22] Yinchuan Li, Xinyu Shao, Jianping Zhang, Haozhi Wang, Leo Maxime Brunswic, Kaiwen Zhou, Jiqian Dong, Kaiyang Guo, Xiu Li, Zhitang Chen, Jun Wang, and Jianye Hao. Generative models in decision making: A survey. In *arXiv preprint arXiv:2502.17100*, 2025.

- [23] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, qiang liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, NIPS*, 2023.
- [24] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *The Thirteenth International Conference on Learning Representations, ICLR*, 2025.
- [25] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, Pieter Abbeel, Jitendra Malik, 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 *Thirty-seventh Conference on Neural Information Processing Systems, NIPS*, 2023.
- [26] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *6th Annual Conference on Robot Learning, CoRL*, 2022.
- [27] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems, RSS*, 2024.
- [28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research, TMLR*, 2024.
- [29] Yatian Pang, Wenzhao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision, ECCV*, 2022.
- [30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *arXiv preprint arXiv:2212.09748*, 2022.
- [31] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems, NIPS*, 2022.
- [32] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. In *arXiv preprint arXiv:2501.15830*, 2025.
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *arXiv preprint arXiv:2103.00020*, 2021.
- [34] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems, NIPS*, 2022.
- [35] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, 2021.
- [36] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, Chelsea Finn, Max Du, Moo Jin Kim, Alexander Khazatsky, Jonathan Heewon Yang, Tony Z. Zhao, Ken Goldberg, Ryan Hoque, Lawrence Yunliang Chen, Simeon Adebola, Gaurav S. Sukhatme, Gautam Salhotra, Shivin Dass,

Lerrel Pinto, Zichen Jeff Cui, Siddhant Haldar, Anant Rai, Nur Muhammad Mahi Shafiullah, Yuke Zhu, Yifeng Zhu, Soroush Nasiriany, Shuran Song, Cheng Chi, Chuer Pan, Wolfram Burgard, Oier Mees, Chenguang Huang, Deepak Pathak, Shikhar Bahl, Russell Mendonca, Gaoyue Zhou, Mohan Kumar Srirama, Sudeep Dasari, Cewu Lu, Hao-Shu Fang, Hongjie Fang, Henrik I Christensen, Masayoshi Tomizuka, Wei Zhan, Mingyu Ding, Chenfeng Xu, Xinghao Zhu, Ran Tian, Youngwoon Lee, Dorsa Sadigh, Yuchen Cui, Suneel Belkhale, Priya Sundaresan, Trevor Darrell, Jitendra Malik, Ilija Radosavovic, Jeannette Bohg, Krishnan Srinivasan, Xiaolong Wang, Nicklas Hansen, Yueh-Hua Wu, Ge Yan, Hao Su, Jiayuan Gu, Xuanlin Li, Niko Suenderhauf, Krishnan Rana, Ben Burgess-Limerick, Federico Ceola, Kento Kawaharazuka, Naoaki Kanazawa, Tatsuya Matsushima, Yutaka Matsuo, Yusuke Iwasawa, Hiroki Furuta, Jihoon Oh, Tatsuya Harada, Takayuki Osa, Yujin Tang, Oliver Kroemer, Mohit Sharma, Kevin Lee Zhang, Beomjoon Kim, Yoonyoung Cho, Junhyek Han, Jaehyung Kim, Joseph J Lim, Edward Johns, Norman Di Palo, Freek Stulp, Antonin Raffin, Samuel Bustamante, João Silvério, Abhishek Padalkar, Jan Peters, Bernhard Schölkopf, Dieter Büchler, Jan Schneider, Simon Guiist, Jiajun Wu, Stephen Tian, Haochen Shi, Yunzhu Li, Yixuan Wang, Mingtong Zhang, Heni Ben Amor, Yifan Zhou, Keyvan Majd, Lionel Ott, Giulio Schiavi, Roberto Martín-Martín, Rutav Shah, Yonatan Bisk, Jeffrey T Bingham, Tianhe Yu, Vidhi Jain, Ted Xiao, Karol Hausman, Christine Chan, Alexander Herzog, Zhuo Xu, Sean Kirmani, Vincent Vanhoucke, Ryan Julian, Lisa Lee, Tianli Ding, Yevgen Chebotar, Jie Tan, Jacky Liang, Igor Mordatch, Kanishka Rao, Yao Lu, Keerthana Gopalakrishnan, Stefan Welker, Nikhil J Joshi, Coline Manon Devin, Alex Irpan, Sherry Moore, Ayzaan Wahid, Jialin Wu, Xi Chen, Paul Wohlhart, Alex Bewley, Wenxuan Zhou, Isabel Leal, Dmitry Kalashnikov, Pannag R Sanketi, Chuyuan Fu, Ying Xu, Sichun Xu, brian ichter, Jasmine Hsu, Peng Xu, Anthony Brohan, Pierre Sermanet, Nicolas Heess, Michael Ahn, Rafael Rafailov, Acorn Pooley, Kendra Byrne, Todor Davchev, Kenneth Oslund, Stefan Schaal, Ajinkya Jain, Keegan Go, Fei Xia, Jonathan Tompson, Travis Armstrong, and Danny Driess. Open x-embodiment: Robotic learning datasets and RT-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition, CoRL*, 2023.

- [37] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning, CoRL*, 2023.
- [38] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2023.
- [39] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, 2023.
- [40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP*, 2020.
- [41] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2024.
- [42] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *arXiv preprint arXiv:2406.09414*, 2024.
- [43] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning, CoRL*, 2019.

- [44] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems, RSS*, 2024.
- [45] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV*, 2023.
- [46] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-VLA: A 3d vision-language-action generative world model. In *Forty-first International Conference on Machine Learning, ICML*, 2024.
- [47] Haoyi Zhu, Honghui Yang, Xiaoyang Wu, Di Huang, Sha Zhang, Xianglong He, Tong He, Hengshuang Zhao, Chunhua Shen, Yu Qiao, and Wanli Ouyang. Ponderv2: Pave the way for 3d foundation model with a universal pre-training paradigm. In *arXiv preprint arXiv:2310.08586*, 2023.
- [48] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, NIPS*, 2024.
- [49] Haoyi Zhu, Honghui Yang, Yating Wang, Jiange Yang, Limin Wang, and Tong He. SPA: 3d spatial-awareness enables effective embodied representation. In *The Thirteenth International Conference on Learning Representations, ICLR*, 2025.

A Details of Experimental Setup

A.1 Policy Network

To evaluate how well Vision Foundation Models (VFs) support advanced Vision-Language Action (VLA) models, we use the RDT [24] architecture as our evaluation policy network, which has demonstrated excellent scalability and strong performance in diffusion-based policy learning. Diffusion timestamps and robot kinematic information are integrated into the policy network using AdaLN-Zero [30]. The vision and language embeddings are used as the Keys and Values in the cross-attention layers to be integrated into the policy network alternately [24]. The Transformer architecture has a hidden dimension of 384, with 6 attention heads, and 12 layers.

For action generation, we use a flow-matching model similar to [3]. Diffusion timestamps are treated as continuous values within the range $[0, 1]$; we do not discretize them. Instead, they are represented using a Fourier embedding with a scale of 0.2 [10]. During training, diffusion timestamps are sampled from a uniform distribution over the interval $[0, 1]$. For inference, we solve the corresponding ODE using the Euler method, dividing the interval $[0, 1]$ into equal-sized steps.

A.2 Details of Benchmarks



Figure 9: **LIBERO simulation benchmark.** We conduct experiments on 40 tasks from four task suites in the LIBERO benchmark. We show two task examples for each suite here.



Figure 10: **MetaWorld simulation benchmark.** We conduct experiments on 30 tasks of three difficulty levels in the MetaWorld benchmark. We show all task examples here.



Figure 11: **Real-world experimental setups.** We conduct experiments on both SO100 and xArm platform. For each robot, we design a suite of 10 tabletop tasks involving diverse objects.

LIBERO. The LIBERO simulation benchmark [23] features a Franka Emika Panda arm in simulation across four challenging task suites: *Goal*, *Spatial*, *Object*, and *Long*. Each suite comprises 10 tasks with 500 demonstrations and is designed to investigate controlled knowledge transfer related to goal variations, spatial configurations, object types, and long-horizon tasks. Unlike prior work [19, 20],

Table 3: Task description of each task in the LIBERO benchmark.

Task Suite	Task Description
LIBERO-Goal	open the middle layer of the drawer put the bowl on the stove put the wine bottle on the top of the drawer open the top layer of the drawer and put the bowl inside put the bowl on the top of the drawer push the plate to the front of the stove put the cream cheese on the bowl turn on the stove put the bowl on the plate put the wine bottle on the rack
LIBERO-Spatial	pick the akita black bowl between the plate and the ramekin and place it on the plate pick the akita black bowl next to the ramekin and place it on the plate pick the akita black bowl from table center and place it on the plate pick the akita black bowl on the cookies box and place it on the plate pick the akita black bowl in the top layer of the wooden cabinet and place it on the plate pick the akita black bowl on the ramekin and place it on the plate pick the akita black bowl next to the cookies box and place it on the plate pick the akita black bowl on the stove and place it on the plate pick the akita black bowl next to the plate and place it on the plate pick the akita black bowl on the wooden cabinet and place it on the plate
LIBERO-Object	pick the alphabet soup and place it in the basket pick the cream cheese and place it in the basket pick the salad dressing and place it in the basket pick the bbq sauce and place it in the basket pick the ketchup and place it in the basket pick the tomato sauce and place it in the basket pick the butter and place it in the basket pick the milk and place it in the basket pick the chocolate pudding and place it in the basket pick the orange juice and place it in the basket
LIBERO-Long	put both the alphabet soup and the tomato sauce in the basket put both the cream cheese box and the butter in the basket turn on the stove and put the moka pot on it put the black bowl in the bottom drawer of the cabinet and close it put the white mug on the left plate and put the yellow and white mug on the right plate pick up the book and place it in the back compartment of the caddy put the white mug on the plate and put the chocolate pudding to the right of the plate put both the alphabet soup and the cream cheese box in the basket put both moka pots on the stove put the yellow and white mug in the microwave and close it

we do not filter out unsuccessful demonstrations, aiming for a more realistic evaluation setting. For policy training, the model predicts action chunks of length 16; after each chunk prediction, 8 steps are executed before generating the next chunk. The observation space includes 2-view RGB images at the current time step, without historical observations. During evaluation, following Liu et al. [23], each task is tested over 50 trials with 3 different random seeds, and success rates are reported. To provide a clearer understanding of the task suites, we present agent-view observations in Figure 9 and detailed task descriptions in Table 3.

MetaWorld. The MetaWorld simulation benchmark [43] includes 50 distinct tabletop manipulation tasks using a Sawyer robot arm. We select 30 tasks from *easy*, *medium*, and *very hard* difficulty levels to evaluate VLA models. We use a scripted policy to collect 20 demonstrations for each task. For policy training, the model predicts action chunks of length 16; after each chunk prediction, 16 steps are executed before generating the next chunk. The observation space consists of a single RGB image at the current time step, without historical observations. During evaluation, each task is tested over 50 trials with 3 different random seeds, and success rates are reported. To better illustrate the task suites, we show agent-view observations in Figure 10 and task descriptions in Table 4.

SO100 Robot Manipulation. The SO100 robot [5] is a low-cost, open-source 6-DoF manipulator, with both the leader and follower arms costing approximately \$250. We assemble the hardware using a 3D-printed kit provided by the open-source community. The robot has two RGB cameras: one mounted on the wrist and the other positioned to provide a third-person view. Both cameras operate at a resolution of 640x480 and 25 FPS. The robot controller runs at 30Hz, and actions are defined as

Table 4: Task description of each task in the MetaWorld benchmark.

Task Name	Task Description
basketball	Dunk the basketball into the basket.
bin-picking	Grasp the puck from one bin and place it into another bin.
button-press	Press a button.
button-press-topdown	Press a button from the top.
button-press-topdown-wall	Bypass a wall and press a button from the top.
button-press-wall	Bypass a wall and press a button.
coffee-button	Push a button on the coffee machine.
coffee-pull	Pull a mug from a coffee machine.
coffee-push	Push a mug under a coffee machine.
dial-turn	Rotate a dial 180 degrees.
disassemble	Pick a nut out of the peg.
door-lock	Lock the door by rotating the lock clockwise.
door-open	Open a door with a revolving joint.
door-unlock	Unlock the door by rotating the lock counter-clockwise.
drawer-close	Push and close a drawer.
drawer-open	Open a drawer.
faucet-close	Rotate the faucet clockwise.
faucet-open	Rotate the faucet counter-clockwise.
hammer	Hammer a screw on the wall.
handle-press	Press a handle down.
handle-press-side	Press a handle down sideways.
handle-pull	Pull a handle up.
handle-pull-side	Pull a handle up sideways.
shelf-place	Pick and place a puck onto a shelf.
soccer	Kick a soccer into the goal.
stick-push	Grasp a stick and push a box using the stick.
sweep	Sweep a puck off the table.
sweep-into	Sweep a puck into a hole.
window-close	Push and close a window.
window-open	Push and open a window.

Table 5: Task description of each task in the SO100 and xArm benchmark. As each parameter combination introduces one task, each task suite has 10 tasks in total. For each task, we test the model for 10 trials.

Task Suite	Task Description	Parameter
SO100	pick [A] and place it on the [B] side of the table	[A]: ["screwdriver", "sponge", "charger"], [B]: ["left", "right"]
	move [A] to the center of the table	[A]: ["cup", "bowl"]
	pick the cloth and wipe the table	None
xArm	unfold the cloth	None
	pick [A] and place it on the [B] side of the table	[A]: ["scissor", "plier", "tap"], [B]: ["left", "right"]
	open the pot lid or put the lid on the pot	[open, close]
	pour the water from the kettle into the cup	None
	place the Moka pot on the cooker	None

target absolute joint angles. Due to its low-cost design, the platform has several hardware limitations, including significant arm jitter, low load capacity, and occasional camera lag, which present practical challenges for developing embodied AI systems. However, given the increasing adoption of such affordable open-source robots by the research community, we believe that evaluating models on these lower-performance systems offers valuable insights and broader applicability. We design four categories of tabletop manipulation tasks for the SO100 setup: (1) **Pick&Place**: involving 3 objects and 2 placement zones (6 tasks), (2) **MoveTo**: navigating 2 objects to a single target zone (2 tasks), (3) **Wipe**: picking up a cloth and wiping the table (1 task), and (4) **Unfold**: unfolding a cloth (1 task). In total, we evaluate performance on 10 distinct tasks. Language instructions for each task are listed in Table 5, and visual examples of the task environments are shown in Figure 11.

During data collection, we record 20 demonstrations per task. For policy training, the model predicts an action chunk of length 64; after each chunk prediction, 40 steps are executed before generating the next chunk. The observation space includes 2-view RGB images at the current time step, along with the absolute joint angles from the current and previous 10 steps. During evaluation, each task is tested over 10 trials, and success rates are reported.

xArm Robot Manipulation. xArm is a high-performance 7-DoF manipulator. The robot is equipped with a third-person view Intel RealSense L515 LiDAR camera, operating at 640×480 resolution and 30 FPS. We collect both RGB and depth images from the camera. The robot controller runs at

30Hz, and actions are defined as target absolute joint angles. We design four categories of tabletop manipulation tasks for the xArm setup: **(1) Pick&Place**: involving 3 objects and 2 placement zones (6 tasks), **(2) Pot**: taking off or putting on the pot lid (2 tasks), **(3) Pour**: pouring water from the kettle into the cup (1 task), and **(4) Moka**: placing the Moka pot on the cooker (1 task). In total, we evaluate performance on 10 distinct tasks. Language instructions for each task are listed in Table 5, and visual examples of task environments are shown in Figure 11.

During data collection, we record 20 demonstrations per task. For policy training, the model predicts an action chunk of length 64; after each chunk prediction, 40 steps are executed before generating the next chunk. The observation space includes a third-person view RGB image at the current time step, as well as the absolute joint angles from the current and previous 10 steps. During evaluation, each task is tested over 10 trials, and success rates are reported.

Table 6: Hyperparameters for EmbodiedMAE training. Since we use pre-training and distillation for different model scales, we use **(P)** to denote pre-training hyperparameters and **(D)** to denote distillation hyperparameters.

Hyperparameters	Values
GPUs	8xNVIDIA L40 (60GB) (P) or 4xNVIDIA Geforce RTX4090 (24GB) (D)
learning rate	3e-4 peak LR (500 steps linear warmup, 300k steps cosine decay to 3e-6)
batch size	512
training steps	200K (P) 100K (D)
input modalities	224x224x3 RGB images, 224x224 Depth maps, 8,192 Point Clouds
image augmentations	ColorJitter(brightness=0.1, contrast=0.1, saturation=0.1, hue=0.05)
trainable parameters	1.1B Giant (P) 304M Large, 87M Base, 22M Small (D) encoders, and 44M decoders
mask ratio	84% (P) 90% (D)
Distillation β	1.0

A.3 DINOv2-RGBD Baseline

To establish a reliable and effective RGBD baseline for practical applications, we follow the approach outlined in Zhu et al. [48], designing a method that naively incorporates depth information based on DINOv2. We introduce an additional Conv2D layer to patchify the depth map, summing the resulting patches with DINOv2’s RGB patchifying output before encoding through the DINOv2 Encoder. We initialize the depth patchifier’s weights and biases to zero, ensuring that the representation model remains functionally equivalent to DINOv2 at the beginning of training. During training, we update only the depth patchifier’s gradients, allowing depth information to be learned following DINOv2’s prior knowledge.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper have discussed the limitations of the work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide every detail to reproduce the main experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will open-source the code in a few days after submission.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide every detail in training and testing.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes. We report error bars in the learning curve.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.).
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Yes. We list the GPU and CPU resources.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Yes, we do.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes, we do.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, they are.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Yes.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not include crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We do not include crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorosity, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.