3D-MVP: 3D Multiview Pretraining for Manipulation

Shengyi Qian[2,*], **Kaichun Mo**[1], **Valts Blukis**[1], **David F. Fouhey**[3], **Dieter Fox**[1], **Ankit Goyal**[1]
[1]NVIDIA [2]University of Michigan [3]New York University
https://jasonqsy.github.io/3DMVP

Abstract: Recent works have shown that visual pretraining on egocentric datasets using masked autoencoders (MAE) can improve generalization for downstream robotics tasks [1, 2]. However, these approaches pretrain only on 2D images, while many robotics applications require 3D scene understanding. In this work, we propose 3D-MVP, a novel approach for 3D multi-view pretraining using masked autoencoders. We leverage Robotic View Transformer (RVT), which uses a multi-view transformer to understand the 3D scene and predict gripper pose actions. We split RVT's multi-view transformer into visual encoder and action decoder, and pretrain its visual encoder using masked autoencoding on large-scale 3D datasets such as Objaverse. We evaluate 3D-MVP on a suite of virtual robot manipulation tasks and demonstrate improved performance over baselines. We also show promising results on a real robot platform with minimal finetuning. Our results suggest that 3D-aware pretraining is a promising approach to improve sample efficiency and generalization of vision-based robotic manipulation policies. We will release code and pretrained models to facilitate future research.

Keywords: Visual pretraining, Manipulation

1 Introduction

Building learning-based manipulation systems is challenging due to the unavailability of diverse large-scale robotics data. To address this, there has been significant interest in using computer vision techniques to learn generalizable visual representations without robotics focused data, for example by self-supervised pre-training on image datasets. In particular, inspired by the success of masked language modeling in NLP, several recent works have explored masked autoencoding (MAE) for visual representation learning [3]. MAE learns to reconstruct randomly masked patches in an input image, encouraging the model to learn high-level semantic features. When applied to egocentric videos from human demonstrations, MAE has been shown to learn representations that generalize well to downstream robotics tasks such as object manipulation [1, 4, 5].

However, current MAE approaches for robotics pretrain only on 2D images, ignoring the 3D structure of the scene. Prior works in robotics have shown that methods that build an explicit 3D visual representation of the environment are more sample efficient and generalize better than those with only 2D visual representations [6, 7, 8]. Hence, in this work, we explore how we can bring the benefits of visual pretraining to robot manipulation methods that reason with explicit 3D representations.

We propose 3D-MVP, a method for 3D Multi-View Pretraining for robot manipulation. Our approach builds upon recent advances in robot manipulation. Specifically, we use the Robotic View Transformer (RVT) [6], a state-of-the-art 3D manipulation method [6]. RVT takes as input a point cloud of the scene and builds a 3D representation of the scene, using a set of fixed orthogonal "virtual" RGBD images. These RGBD images are fed through a transformer model that fuses information across views and predicts robot actions in the form of future gripper poses.

We choose RVT over other methods for manipulation that build a 3D representation of the scene (e.g. PerAct [7] and Act3D [9]), because other methods use either voxels or point clouds as input

[*] The work was done during a NVIDIA internship.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

a transformer model, while RVT uses orthogonal RGBD images. The view-based representation makes RVT a suitable candidate for MAE pretraining.

We pretrain the multi-view representation in RVT by attaching it to a lightweight MAE decoder. We then randomly mask out a subset of the visual tokens for each view and train the model to reconstruct the multiview RGB-D images. After the pre-training, we discard the decoder. We then fine-tune the visual encoder along with RVT's action decoder on various manipulation tasks.

In order to learn generalizable and robust visual features, we use the recent works that have led to the creation of large-scale datasets of 3D scenes, such as Objaverse and 3D-FRONT [10, 11, 12, 13]. These datasets contain high-quality 3D scans of indoor environments along with realistic textures and materials. We use these datasets to create sets of orthogonal views that are similar to the 3D representation used in RVT. These sets of orthogonal views are then used for pretraining the visual encoder in RVT. We conduct experiments to ablate various different choices available for pre-training. Specifically, we study how masking strategies, dataset choice and dataset sizes affect the downstream manipulation performance.

Finally, we evaluate 3D-MVP on the RLBench benchmark [14], a suite of manipulation tasks in a simulated environment. We find that pretraining the RVT encoder with 3D-MVP leads to significant improvements over training from scratch or pretraining with 2D MAE. These results inform how we can advance the state-of-the-art in robotic manipulation with the help of pretraining. We further evaluate 3D-MVP on the Colosseum benchmark [8], which tests a system's generalization across various unseen variations of manipulation tasks like object size change, color change, and lighting changes. We find that the proposed 3D-MVP method is more robust across various variations than RVT trained from scratch.

In summary, our contributions are three-fold.

• We propose 3D-MVP, a novel approach for 3D multi-view pretraining using masked autoencoding on large-scale 3D datasets.

• We study how various design choices in pretraining, like masking strategy, dataset combination and sizes, affect downstream object manipulation performance.

• We demonstrate that pretraining with 3D-MVP leads to significant improvements on object manipulation tasks. We also show that 3D-MVP enables training policies that are more robust to variations such as size, texture, and lightning, on the COLOSSEUM benchmark.

2 Related Work

Our work builds upon several active areas of research, including self-supervised learning, visual pretraining for robotics, and learning robotic manipulation from demonstrations.

Self-supervised learning. Self-supervised learning aims to learn useful representations from unlabeled data by solving pretext tasks that do not require manual annotation. Early work in this area focused on designing pretext tasks for 2D images, such as solving jagsaw puzzles [15], constrastive learning [16, 17] or joint embedding approaches [18, 19, 20, 21, 22, 23]. Most related to our work is the masked autoencoder (MAE) approach proposed by He et al. [3], which learns to reconstruct randomly masked patches in an image. MAE has been shown to learn transferable representations for object detection and segmentation tasks. Furthermore, Bachmann et al demonstrates MAE pre-training can be extended to different modalities such as semantics and depth [24]. In this work, we extend the MAE approach to multi-view 3D scenes, enabling us to learn 3D-aware representations that are useful for robotic manipulation tasks.

Visual pretraining for Robotics. Visual pretraining has demonstrated impressive generalization ability on computer vision tasks. Therefore, prior works have explored whether it works for robotics tasks as well. Specifically, the robotics community has trended towards learning representations using state-of-the-art self-supervised vision algorithms on diverse interaction datasets [25, 26, 27], and finetune the network on robotics tasks [28, 2, 1, 4, 29, 5]. 3D-MVP follows the same procedure. However, existing robotics pretraining approaches typically learn a 2D visual encoder (e.g. ResNet [30] or ViT [31]), we find they are inferior than manipulation policies which do explicit 3D modeling (e.g. RVT [6], Act3D [9]). Migrating a pretrained ViT to 3D manipulation policies is nontrivial since they do not have a 2D visual encoder. In this paper, we propose 3D-MVP, which does 3D-aware pretraining on 3D manipulation policies, to fill the gap.

Learning manipulation from demonstrations. Recent work has explored using transformers for multi-task manipulation policies that predict robot actions from visual and language inputs [7, 32, 33, 34, 35]. End-to-end models like RT-1 [36], GATO [37], and InstructRL [33] directly predict 6-DoF end-effector poses but require many demonstrations to learn spatial reasoning and generalize to new scenes. To better handle 3D scenes, PerAct [7] and C2F-ARM [38] voxelize the workspace and detect the 3D voxel containing the next end-effector pose. However, precise pose prediction requires high-resolution voxels which are computationally expensive. Recently, RVT [6] proposes a multi-view transformer that attends over point cloud features from multiple camera views to predict actions. This avoids explicit voxelization and enables faster training and inference than PerAct. Act3D [9] represents the scene as a continuous 3D feature field and samples points to featurize with attention, allowing adaptive resolution. GNFactor [39] jointly optimizes a generalizable neural field for reconstruction and a Perceiver for decision-making. In contrast, our proposed 3D-MVP learns 3D scene representations through masked autoencoding pretraining on a large dataset of 3D object models. This pretraining enables 3D-MVP to build a rich understanding of 3D geometry and semantics prior to finetuning on downstream manipulation tasks.

3 Approach

In this section we first provide essential background on RVT, then define our method 3D-MVP that learns 3D-aware representations for robotic manipulation using masked autoencoding on multi-view 3D scenes, and finally describe how we finetune the method on downstream manipulation tasks. Figure 1 gives an overview of our approach.

3.1 Background on Robotic View Transformer (RVT).

It is a state-of-the-art object manipulation method [6]. It creates an explicit 3D representation of the scene by using orthogonal virtual views of a scene. Please refer to Goyal et al. [6] for a full explanation. Here, we provide a brief overview and define the notation.

RVT takes a point cloud of the robot workspace as input (Fig. 1). RVT is agnostic to the poses of the RGBD cameras used to construct the input point cloud. For example, it can be obtained from a combination of third-person cameras around the workspace, head cameras, or wrist cameras. RVT then renders this point cloud using a set of five "virtual" cameras placed at orthogonal locations around the robot. The virtual cameras are placed at the top, left, right, front, and back of the robot workspace with respect to the robot. Each virtual image has 10 channels: RGB (3 channels), Depth (1 channel), 3D point coordinate in world frame (3 channels), and 3D point coordinate in camera sensor frame (3 channels). We denote the virtual images captured from different virtual camera poses $\{p_1, \ldots, p_5\}$ as $\{I_1, \ldots, I_5\}$.

These virtual images are then tokenized into N patch embeddings [31], flattened into a sequence of $5N$ tokens spanning all images, and fed to a multi-view transformer. The goal of RVT's multi-view transformer is to learn a function f_θ that maps the virtual images as well as language instructions L to the 6-DoF end-effector pose and the gripper's binary open or close state:

$$a_{\text{pos}}, a_{\text{rot}}, a_{\text{open}} = f_\theta(L, I_1, p_1, \ldots, I_5, p_5) \tag{1}$$

RVT is trained end-to-end from scratch on sampled trajectories from simulator or real robots. While RVT has shown state-of-the-art results on 3D manipulation, it does not generalize to novel objects and scenes. In the next section, we describe our novel approach 3D-MVP, and how we modify and pretrain the RVT encoder using 3D-MVP.

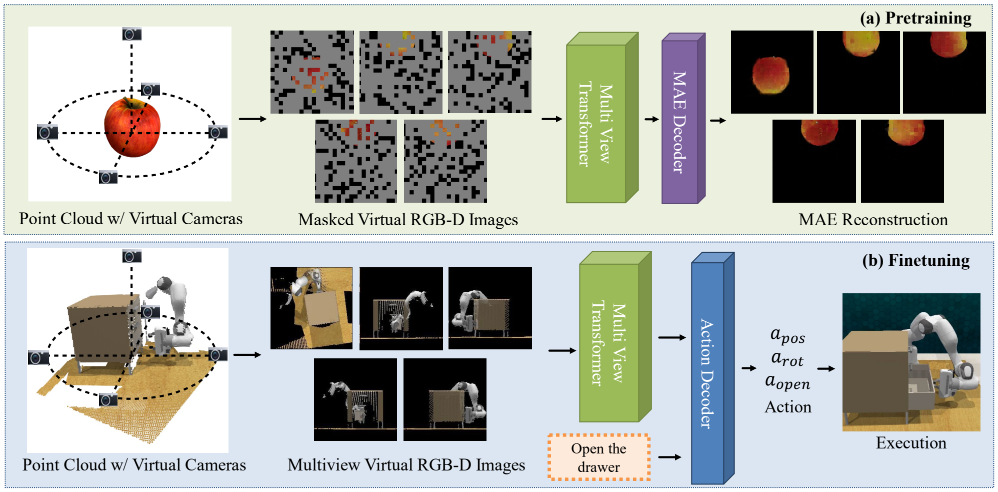

Figure 1: Overview of 3D-MVP. (a) We first pretrain a Multiview 3D Transformer using masked autoencoder on multiview RGB-D images. (b) We then finetune the pretrained Multiview 3D Transformer on manipulation tasks. Since the MVT is pretrained, the learned manipulation policy generalizes better. For example, it is more robust to changes of texture, size and lighting.

3.2 3D Multi-View Pretraining (3D-MVP)

Architecture change to RVT. The key idea is to pretrain the RVT visual encoder f_θ using masked autoencoding on large-scale 3D scene datasets. However, RVT's multiview transformer f_θ is an end-to-end model that takes language instructions as input, and produces robot actions. Existing robotics data with language and actions is limited in terms of diversity of 3D scenes, and 3D scene datasets do not typically contain robotics annotations.

To enable pre-training on 3D scene datasets, we first split the multiview transformer f_θ into an input renderer $\mathcal{R}$, an encoder network $\mathcal{E}$ and an action decoder network $\mathcal{D}$. The renderer $\mathcal{R}$ maps the posed input images into the five virtual images, by constructing a point cloud and rendering it from the five views:

$$\{I_1, \ldots, I_5\} = \mathcal{R}(I_1, p_1, \ldots, I_5, p_5) \tag{2}$$

The encoder $\mathcal{E}$ maps the virtual images into a latent embedding $z \in \mathbb{R}^{5N \times H}$ (where H is the hidden size) and the action decoder $\mathcal{D}$ maps z to the robotic action space, i.e.,

$$a_{\text{pos}}, a_{\text{rot}}, a_{\text{open}} = \mathcal{D}(L, z), \quad z = \mathcal{E}(I_1, \ldots, I_5)\ , \tag{3}$$

where tokenization of the virtual images into $5N$ patch embeddings is subsumed into $\mathcal{E}$. Both encoder $\mathcal{E}$ and decoder $\mathcal{D}$ are multiview transformers. We keep the decoder lightweight to focus on pretraining of the encoder.

Pretraining encoder $\mathcal{E}$. Our visual pretraining focuses on learning a generalizable representation for the encoder $\mathcal{E}$. We extract point clouds from Objaverse and render the point cloud using the same five "virtual" cameras. Given 5 virtual images $\{I_1, \ldots, I_5\}$, we randomly mask out a subset of the visual tokens for each view, and denote the masked inputs as $\{I'_1, \ldots, I'_5\}$. We use the encoder to extract the embedding z from the masked inputs,

$$z = \mathcal{E}(\{I'_1, \ldots, I'_5\}) \tag{4}$$

We use a separate, lightweight MAE decoder $\mathcal{D}_{MAE}$ to reconstruct the original image $\{I_1, \ldots, I_5\}$ from the embedding z.

$$\{\tilde{I}_1, \ldots, \tilde{I}_5\} = \mathcal{D}_{MAE}(z) \tag{5}$$

The encoder $\mathcal{E}$ and decoder $\mathcal{D}_{MAE}$ are trained end-to-end using a pixel-wise reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \frac{1}{5WH} \sum_{i=1}^{5} \sum_{p=1}^{W \cdot H} \|[I_i]_{(p)} - [\tilde{I}_i]_{(p)}\|_2^2\ , \tag{6}$$

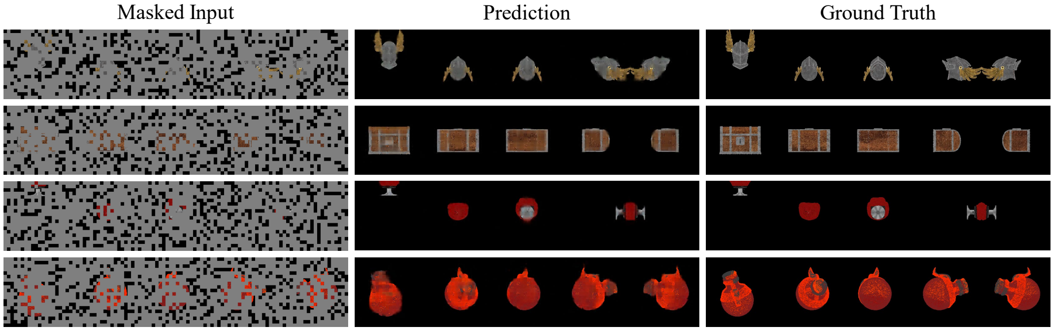

Figure 2: MAE Reconstruction results on Objaverse. Our pretrained multi-view transformer generalizes to unseen object instances and reconstructs multi-view images from their masked versions.

where $[I]_{(p)}$ indexes the image $I \in \mathbb{R}^{W \times H \times C}$ at pixel p. By jointly learning to reconstruct all five images and varying the masking patterns during training, we hypothesize that the encoder will learn to reason across the multiple views and extract 3D-aware features that are robust to occlusions and viewpoint changes. In order to inform future works, we study how various masking strategies and dataset combinations affect the final downstream performance (See Tab. 2).

Implementation details. We implement the pretraining using the PyTorch library and train it on 8 NVIDIA V100s. We sample 200K high-quality 3D models with realistic textures and materials from Objaverse [10] for pretraining. We use a patch size of 10x10 to tokenize images. For the encoder $\mathcal{E}$, we use a multiview transformer with 8 layers, 8 attention heads and a hidden dimension of 1024. For the decoder, we use a multiview transformer with 2 layers and 8 attention heads. We train the model for 15 epochs using the AdamW optimizer with a learning rate of 0.0001 and a weight decay of 0.01. We use a batch size of 3 and a masking probability of 0.75.

3.3 Finetuning on Downstream Manipulation Tasks

To adapt the pretrained encoder $\mathcal{E}$ for a specific manipulation task, we finetune it along with the action decoder $\mathcal{D}$ on a dataset of manipulation demonstrations. Assume a demonstration consists of tuples of virtual images, language goals and actions. During training, we randomly sample a tuple and supervise the network to predict the action given the virtual images and the language goal.

Implementation details. For finetuning on manipulation demonstrations, we follow the standard practice [6, 8]. We use 8 NVIDIA V100 (32GB) for finetuning and a single V100 for evaluation. The learning rate is 1e-4 with warmup. We use Lamb [40] as the optimizer and the batch size is 3.

4 Experiments

In this section, we evaluate the effectiveness of 3D-MVP for robotic manipulation tasks, and aim to answer the following questions: (1) Does 3D-aware pretraining improve manipulation performance compared to training from scratch or 2D pretraining? (2) Does 3D-aware pretraining improve robustness to environmental variations encountered in manipulation tasks? (3) How do various design choices while pretraining affect the downstream manipulation performance? To answer these questions, we evaluate 3D-MVP on two benchmarks: RLBench [14] for general manipulation performance and COLOSSEUM [8] for systematic evaluation of robustness to environmental perturbations.

4.1 Validating 3D Masked Autoencoding

We validate whether masked autoencoder works in our setup with multi-view images from 3D assets. Specifically, we check whether the pretrained multi-view transformer generalizes to unseen 3D assets from Objaverse [10]. We validate it qualitatively in Figure 2. We find that the pretrained 3D-MVP network achieves high-fidelity reconstructions despite 75% of the input points being masked, suggesting that 3D-MVP learns meaningful 3D representations through this pretext task.

Models	Average Success	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Open Drawer	Place Cups	Place Wine	Push Buttons	Put in Cupboard	Put in Drawer	Put in Safe	Screw Bulb	Slide Block	Sort Shape	Stack Blocks	Stack Cups	Sweep to Dustpan	Turn Tap
Image-BC (CNN) [42, 7]	1.3	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	8.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0
Image-BC (ViT) [42, 7]	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.0
C2F-ARM-BC [38, 7]	20.1	24.0	24.0	4.0	20.0	20.0	0.0	8.0	72.0	0.0	4.0	12.0	8.0	16.0	8.0	0.0	0.0	0.0	68.0
PerAct [7]	49.4	55.2	89.6	5.6	70.4	**88.0**	2.4	44.8	92.8	28.0	51.2	84.0	17.6	74.0	16.8	26.4	2.4	52.0	88.0
RVT [6]	62.9	52.0	99.2	11.2	88.0	71.2	**4.0**	91.0	**100**	49.6	88.0	91.2	48.0	**81.6**	**36.0**	28.8	26.4	72.0	93.6
3D-MVP (Ours)	**67.5**	**76.0**	**100**	**20.0**	**96.0**	84.0	**4.0**	**100**	96.0	**60.0**	**100.0**	**92.0**	**60.0**	48.0	28.0	**40.0**	**36.0**	**80.0**	**96.0**

Table 1: Results on RLBench [14]. We report the task completion success rate for 18 RLBench tasks, as well as the average success rate. 3D-MVP reaches the state-of-the-art performance on the benchmark. The pretraining is mainly helpful for tasks with medium difficulty.

4.2 Results on RLBench

Setup. RLBench [14] is a popular simulation benchmark for learning manipulation policies. Each task requires the robot to perform a specific action, such as picking up an object, opening a drawer, or stacking blocks. We follow the simulation setup of PerAct [7] and RVT [6] and use CoppelaSim [41] to simulate 18 RLBench tasks. A Franka Panda robot with a parallel gripper is controlled to complete the tasks. The 18 RLBench tasks are the same as PerAct and RVT. The visual observations are captured from four noiseless RGB-D cameras positioned at the front, left shoulder, right shoulder, and wrist with a resolution of 128×128.

Baselines. We compare 3D-MVP with the following baselines on RLBench: (1) **Image-BC** [42] is an image-to-action behavior cloning approach which takes the visual observation and predict the corresponding action. We compare with two variants which use CNN and ViT as the visual encoders, and call them Image-BC (CNN) and Image-BC (ViT), respectively; (2) **C2F-ARM-BC** [38] is another behavior cloning approach which converts RGB-D obversations to multi-resolution voxels and predicts the next key-frame action. (3) **PerAct** [7]: a multi-task a Perceiver transformer for robotic manipulation. The inputs are point clouds with color features and PerAct uses a Perceiver IO network to compress them to a fixed dimension [43]. (4) **RVT** [6]: The same Robotic View Transformer architecture as 3D-MVP but trained from scratch on the downstream tasks. We do not compare with 2D pretraining methods since they do not work well on RLBench [8].

Metrics. We report the task success rate for each individual tasks, and the average success rate.

Results. We show quantitative results on Table 1. For the average success rate, 3D-MVP outperforms existing state-of-the-art methods, PerAct [7] and RVT [6], by a large margin. It demonstrates the effectiveness of bringing visual pretraining for manipulation policy which has explicit 3D modeling. We find the improvement of 3D-MVP mainly comes from tasks which has medium difficulty, such as "insert peg", "put in cupboard", and "stack blocks". If the task is too hard and PerAct/RVT is not able to solve any of them, the pretraining does not help. If the task is too easy and PerAct/RVT has already reached >90 success rate, the pretraining has limited space of improvement.

4.3 Results on COLOSSEUM

We further evaluate its generalization ability and robustness to environmental variations.

Benchmark. COLOSSEUM [8] is a benchmark for evaluating generalization for robotic manipulation. It contains 20 different tasks such as hockey, empty dishwasher. For each task, it has 12 environmental perturbations, including changes in color, texture, size of objects and backgrounds, and lightnings. The objects which can be changed include Manipulation object (`MO`), Receiver Object

(RO) and the table. Therefore, it is well-suited for evaluating the generalization ability of manipulation approaches with pretraining.

Simulation setup. For simulation, we follow the original COLOSSEUM setup. We use CopppelaSim [41] to simulate all tasks. In training, we do not add any environmental perturbations and generate 100 demonstrations for each task. During test time, we generate 12 environmental perturbations for each task and 25 demonstrations for each perturbation. In some cases, it is not possible to generate a plausible perturbation for some scenarios. For instance, it is hard to perturb the environment correctly for "empty dishwashers". Therefore, we only report results on settings we are able to generate. We report baselines on exactly the same setting for fair comparison.

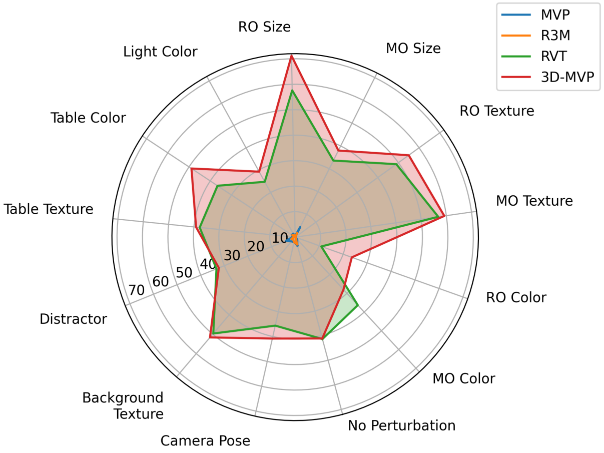

Figure 3: Results on COLOSSEUM [8]. We report the average task completion success rate for 12 environmental perturbations and no perturbation. Manipulation policies which do explicit 3D reasoning (RVT [6] works significantly better and 2D pretraining approaches (MVP [1] and R3M [2]). 3D-MVP is more robust than RVT on most perturbations. MO = manipulation object. RO = receiver object.

Metrics. We also report the task completion success rate on COLOSSEUM. Instead of averaging for each task, we report the average success rate for each environmental perturbation.

Baselines. We compare 3D-MVP with state-of-the-art baselines reported on COLOSSEUM, which includes RVT and two 2D pretraining approaches: (1) MVP [1, 4]: a 2D pretraining approach using MAE reconstruction losses. It is pretrained on a collection of interaction datasets, such as Ego4D [25], EpicKitchen [27], and 100DOH [26]. The pretrained encoder is then finetuned and evaluated on COLOSSEUM. (2) R3M [2]: a 2D pretraining approach using a combination of reconstruction and contrastive losses. It is pretrained on Ego4D [25]. The pretrained encoder is then finetuned and evaluated on COLOSSEUM. (3) RVT [6]: Trained on COLOSSEUM from scratch.

Results. We show results in Figure 3. First, our method outperforms existing 2D pretraining (MVP [1], and R3M) significantly. It indicates existing 2D pretrainig methods are not ready for complicated robotic manipulation. Compared with RVT which is trained from scratch, our method is more robust to most perturbations. It is especially robust to the change of texture and size of Receiver Object (RO), size of the manipulation object (MO), Light color and Table color. We believe it is because the pretraining stage enables our approach to see diverse 3D objects.

4.4 Ablation Studies

To analyze the impact of different design choices in 3D-MVP, we conduct ablation studies on the RLBench benchmark. Table 2 shows the average success rates of 3D-MVP with different network architecture, masking strategies and pretraining datasets. And we discuss results as follows.

Network architecture change. We validate whether the performance boost comes from architecture change in Sec. 3.2 or the pretraining. we finetune our approach without pretraining and find the performance is 62.9, similar to the original RVT [6].

Should we pretrain on object or room-level data? The choice of pretraining datasets are typically critical for self-supervised learning. In our experiments, we mainly use Objaverse [10], which is a object-centric 3D datasets. Since we are mainly evaluated on tabletop manipulation, we also try

Network Architecture	Pretraining Datasets	Masking Strategy	Success Rate
3D-MVP	Objaverse (full) [10]	RGB	67.6
3D-MVP	Objaverse (small) [10]	RGB	65.3
3D-MVP	Objaverse (full) [10]	All	64.4
3D-MVP	3D-FRONT [11]	RGB	63.6
3D-MVP	RLBench [14]	RGB	67.5
3D-MVP	RLBench [14]	All	64.7
3D-MVP	None	None	62.9
RVT [6]	None	None	62.9

Table 2: Ablation studies on the RLBench benchmark. We analyze the contribution of our network architecture, pretraining datasets, and the masking stretegy. For each variant, we report the average task completion success rate on RLBench [14].

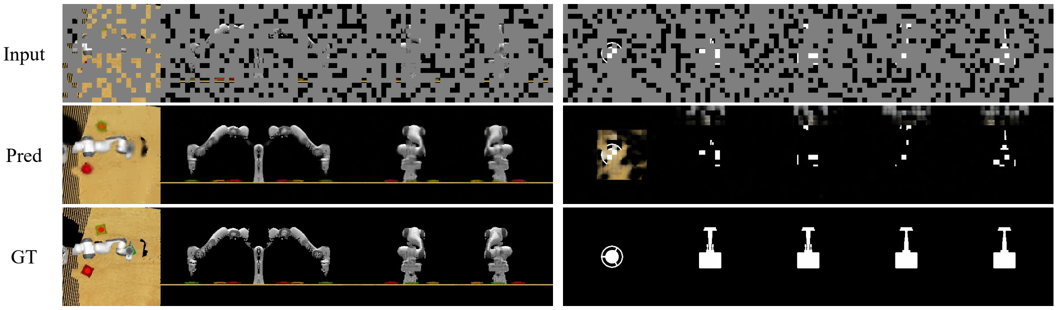

Figure 4: Pretraining MAE on RLBench scenes leads poor generalization performance. **(Left):** MAE reconstruction results on unseen RLBench renderings. **(Right)**: MAE reconstruction results on Objaverse renderings. While the reconstruction is reasonable on RLBench unseen renderings, it overfits to RLBench and does not learn a general representation.

room-level 3D datasets such as 3D-FRONT [11]. We conduct the experiments on RLBench, and observe pretraining on 3D-FRONT only boosts the performance mildly from 62.9 to 63.6. In comparison, pretraining on Objaverse boosts the performance to 67.6. This suggests that the diversity and scale of the pretraining data are important for learning generalizable representations.

Does more pretraining data help? We sampled 18K objects from Objaverse and called it Objaverse (small). The full Objaverse dataset we use has 200K objects. When we pretrain the encoder with Objaverse (small), we get a 65.3 mean success rate, which is worse than using Objaverse (full). This suggests that a larger dataset of pretraining helps performance in the downstream task.

Can we pretrain on RLBench? Instead of pretraining on Objaverse, we investigate whether we can just pretrain on RLBench point clouds. We find it can achieve an average success rate of 67.5 on RLBench test set, which is comparable to Objaverse. However, the pretrained model does not generalize as shown in Figure 4.

Masking strategy. We also observe that masking RGB channels performs better than masking all channels. We hypothesize that the network finds it very challenging to reconstruct all the channels and is unable to learn better visual representations. We view our findings as similar to He et al. [3], who find that the benefits of pretraining diminish if the pretraining task becomes too difficult, like when the masking ratio becomes very high (>80%).

5 Conclusion

In this work, we introduced 3D-MVP, a novel approach for 3D multi-view pretraining using masked autoencoders to improve the performance and generalization of manipulation policies. Future works include extending 3D-MVP to generalize to novel manipulation tasks.

Acknowledgment. We thank Ishika Singh and Jiafei Duan for their help with the COLOSSEUM benchmark.

References

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

[2] 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.

[3] 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*, pages 16000–16009, 2022.

[4] 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*, pages 416–426. PMLR, 2023.

[5] S. Dasari, M. K. Srirama, U. Jain, and A. Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*, 2023.

[6] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

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

[8] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024.

[9] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

[10] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[11] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[12] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024.

[13] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

[14] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[15] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[18] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision*, pages 456–473. Springer, 2022.

[19] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[20] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[21] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[22] 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*, 33: 21271–21284, 2020.

[23] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[24] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022.

[25] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[26] D. Shan, J. Geng, M. Shu, and D. Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.

[27] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

[28] 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. *arXiv preprint arXiv:2210.00030*, 2022.

[29] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36, 2024.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[32] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, pages 175–187. PMLR, 2023.

[33] H. Liu, L. Lee, K. Lee, and P. Abbeel. Instruction-following agents with jointly pre-trained vision-language models. 2022.

[34] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[35] A. Simeonov, A. Goyal, L. Manuelli, L. Yen-Chen, A. Sarmiento, A. Rodriguez, P. Agrawal, and D. Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. *arXiv preprint arXiv:2307.04751*, 2023.

[36] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[37] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[38] S. James, K. Wada, T. Laidlow, and A. J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *CVPR*, 2022.

[39] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023.

[40] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

[41] E. Rohmer, S. P. Singh, and M. Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 1321–1326. IEEE, 2013.

[42] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[43] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.