Investigate Transfer Learning For Pre-Trained Visual Foundation Encoder on Robot Manipulation Policy

Yuwei Lin Wei Lin Pai Yuchi Hsu
Stanford University
{ywlin, wpai, yuchihsu}@stanford.edu

Abstract

Robotic manipulation represents a fundamental challenge in robotics research, requiring sophisticated visual-motor policies that can interpret complex 3D scenes and execute precise actions. While diffusion-based policy learning has demonstrated promising capabilities in generating coherent, multi-modal action trajectories for robotic manipulation tasks, these approaches suffer from critical limitations in sample efficiency and cross-domain generalization. Current methods typically require extensive task-specific demonstration data and struggle to transfer knowledge to novel objects or environmental configurations, significantly constraining their practical deployment in real-world settings.

Our work explores the potential of integrating internet-scale 3D foundation models such as Uni3D [14] with diffusion policy networks through parameter-efficient adaptation techniques. We investigate whether leveraging pre-trained 3D representations, which encode rich semantic and spatial understanding from diverse visual data, can improve sample efficiency and enable better generalization to unseen manipulation scenarios while maintaining the expressive power of diffusion-based action generation. Through preliminary experiments across manipulation benchmarks, we examine the challenges and opportunities in bridging foundation models with behavioral policies. Our findings provide insights into the integration of large-scale pre-trained representations with robot learning, revealing both the potential benefits and current limitations of this approach for scalable robotic manipulation.

1. Introduction

Visual-motor policies for robotics requires leveraging high-level perceptual features for long-horizon planning while maintaining the flexibility to generalize across diverse and novel tasks. Traditional imitation learning approaches often struggle with the complexity of real-world scenarios, where task complexity demand robust representation learning capabilities. Recently, diffusion-based policy learning has emerged as a promising framework for robotic control, generating coherent multi-step action sequences through iterative denoising processes that naturally handle the inherent stochasticity and multi-modality present in manipulation tasks. However, diffusion-based policies typically require prohibitively large amounts of high-quality demonstration data while exhibiting limited generalization to out-of-domain tasks and novel environmental conditions. The data hunger of these approaches becomes particularly problematic when scaling to diverse manipulation scenarios, as collecting sufficient demonstrations for each new domain or task configuration remains both time-consuming and resource-intensive.

Simultaneously, the emergence of 3D foundation models has revolutionized spatial understanding by providing rich geometric representations learned from large-scale 3D data. Uni3D [14] represents a significant advancement in this direction, presenting a unified and scalable 3D pre-training framework that uses a 2D initialized Vision Transformer (ViT) pretrained end-to-end to align 3D point cloud features with image-text aligned features. These models demonstrate exceptional capabilities across diverse 3D tasks, showcasing their ability to understand complex 3D geometric structures and semantic relationships without task-specific fine-tuning.

To address the generalization limitations of diffusion-based policies, we propose leveraging the rich 3D geometric representations from foundation models like Uni3D to enhance policy learning through transfer learning techniques. Our key hypothesis is that the pre-trained 3D features, which already encode robust spatial and semantic understanding from large-scale data, can be effectively transferred to manipulation tasks through parameter-efficient adaptation methods. By integrating these foundational 3D representations with diffusion policy networks via lightweight adaptation modules, we aim to significantly reduce the data requirements for policy training while improving generalization to novel objects, scenes, and manipula-

tion scenarios that were not present in the original demonstration dataset.

2. Related Work

Robotic manipulation requires translating rich perceptual inputs into precise and robust control actions. Recent work [11] has focused on leveraging pretrained vision models to reduce training costs and improve generalization. Our approach builds on recent innovations in this direction, aiming to improve both parameter efficiency and policy expressiveness through modular design. Below, we review several key works that have shaped the motivation and methodology of our project.

2.1. Lossless Adaptation of Pretrained Vision Models for Robotic Manipulation

The core contribution of [11] is a "lossless adaptation" framework that enables task-specific learning by inserting lightweight adapter modules into frozen, pretrained vision encoders. This approach avoids modifying the original encoder weights, requiring less than 5% additional parameters while recovering over 95% of the performance gap between frozen-feature baselines and fully fine-tuned models.

In this framework, RGB images are first processed by a variety of pretrained visual backbones, including both supervised models (e.g., ViTs [3], NFNets [1], ResNets [6]) and self-supervised models (e.g., CLIP [10], BYOL[4], Visual MAE [5]). The resulting visual features are then passed through compact adapter modules before being fed into a linear policy head that outputs the robotic arm's actions. The entire system is trained using supervised imitation learning, relying on expert demonstrations to map observations to actions.

While this research demonstrates the effectiveness of this paradigm, it does not incorporate several recent advances in action modeling and visual representation. In our work, we seek to extend this paradigm by integrating state-of-the-art training techniques such as Diffusion Policy [2], as well as more expressive visual encoders like Uni3D [14].

2.2. Diffusion Policy

Diffusion Policy [2] models robot control as a conditional denoising diffusion process over action sequences, learning to iteratively refine Gaussian noise into precise robot actions. Compared to energy-based policies, diffusion policy avoids costly normalization, enables receding-horizon control, and cleanly decouples perception from action.

In the experiments, diffusion policy delivers a 46.9% improvement over prior behavior-cloning baselines on twelve simulated and real-robot tasks (2–6 DoF, rigid and fluid interactions), outperforming mixture-of-Gaussians and implicit energy-based policies while maintaining real-time inference with a modest number of denoising steps.

3D Diffusion Policy (DP3) extends this paradigm by applying diffusion on compact 3D embeddings, produced by an efficient MLP encoder from sparse point clouds, as well as robot poses. This 3D modality yields remarkable data efficiency and a 24.2% average performance gain over the baseline, while also improving generalization to various spatial configurations, viewpoints, and object instances and reducing erratic actions in safety-critical settings.

While DP3 takes an important step by introducing 3D perception into the diffusion framework, its encoder remains relatively shallow and lacks the capacity for strong zero-shot generalization. To further strengthen the perceptual component of this pipeline, we turn to recent advances in scalable 3D representation learning.

2.3. Uni3d: Exploring unified 3d representation at scale

Uni3D [14] makes a significant leap in 3D representation learning by expanding the standard Vision Transformer architecture into 3D. Specifically, it converts point clouds into patch tokens and scaling it up to one billion parameters. It achieves strong zero-shot classification and transfers directly to part segmentation and real-scene recognition tasks like ScanNet without additional real-world data.

This unified model also supports applications such as shape retrieval and 3D "painting" without any fine-tuning, demonstrating broad versatility across downstream 3D tasks.

The above literature informs a natural synthesis: combining the lossless adaptation framework with the powerful planning capabilities of diffusion models, and enhancing both through strong 3D encoders like Uni3D. Our work explores this integration, aiming to build a vision-conditioned policy that is modular, scalable, and efficient in both data and parameters.

3. Methods

This section presents our approach for integrating pretrained 3D vision representations with diffusion-based policy learning, followed by a description of the baseline method used for comparison.

3.1. U3DP: Uni3D-Enhanced 3D Diffusion Policy

We propose U3DP (Unified 3D Diffusion Policy), which builds upon the 3D Diffusion Policy (DP3) framework by replacing its simple point cloud encoder with the pre-trained Uni3D model. The resulting architecture consists of a Uni3D perception module and a diffusion policy module.

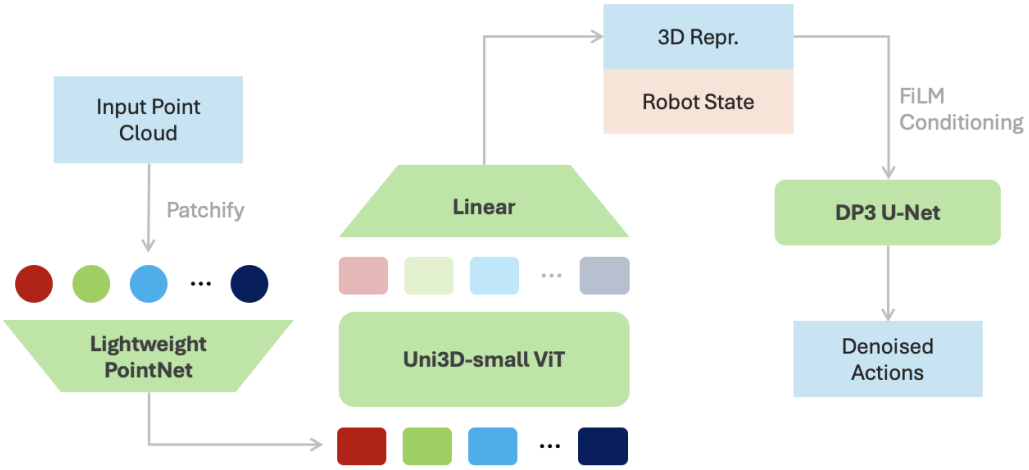


Figure 1. Architecture of U3DP. The input point cloud is first processed through point tokenization, where points are grouped into local patches and processed by a lightweight PointNet to extract patch-level features. These features are then encoded by the pre-trained Uni3D-Small Vision Transformer (ViT) to produce semantically rich 3D representations. A learnable linear projection layer maps the 1024-dimensional Uni3D features to 64-dimensional vectors to balance with the 64-dimensional robot state features. The concatenated visual and proprioceptive features are integrated into the DP3 U-Net denoising network via Feature-wise Linear Modulation (FiLM) conditioning to generate denoised action sequences through the diffusion process.

3.1.1 Perception Module

Instead of using the original DP3's lightweight MLP encoder, we employ the pre-trained Uni3D-Small model, which contains approximately 23 million parameters. The Uni3D model is a Vision Transformer (ViT) that has been pre-trained on approximately one million 3D shapes paired with 10 million images and 70 million text descriptions, enabling it to extract semantically rich 3D representations.

Point Cloud Preprocessing. Before feeding the point cloud into Uni3D, we apply a specific normalization strategy that aligns with Uni3D's pre-training data distribution. Unlike DP3's original normalization that scales coordinates to $[-1, 1]$, we adopt Uni3D's normalization scheme: First, we center the point cloud by subtracting its centroid. Then, we scale the coordinates by dividing by the furthest distance from the centroid, ensuring spatial coordinates lie within a unit sphere. Color values are normalized to $[0, 1]$ by dividing by 255. This normalization strategy ensures that our fine-tuning data distribution matches that of Uni3D's pre-training phase, facilitating better transfer learning.

Point Tokenization. The normalized point cloud is processed through Uni3D's point tokenizer, which follows the PointBERT[13] architecture. The tokenizer first groups points into local patches using FPS to select center points, followed by k-nearest neighbor (kNN) clustering. Each local patch is then processed through a lightweight PointNet[9] to extract patch-level features, resulting in a sequence of tokens that can be processed by the transformer architecture.

Feature Projection. The Uni3D-Small model outputs 384-dimensional features through its transformer backbone, which are then projected to 1024 dimensions as part of Uni3D's standard architecture. To integrate these features with DP3's diffusion policy module, we introduce an additional learnable linear projection layer that maps the 1024-dimensional features to 64-dimensional vectors. This dimensional reduction is crucial for maintaining balance between the visual and proprioceptive modalities: the robot pose features are also 64-dimensional, and both are concatenated before being fed into the U-Net. Without this projection, the high-dimensional visual features (1024) would dominate the lower-dimensional pose features (64) during the diffusion process, potentially leading to suboptimal policy learning where proprioceptive information is underutilized.

3.1.2 Diffusion Policy Module

The diffusion policy module follows the same structure as in DP3, utilizing a U-Net architecture as the denoising network backbone. Given the encoded observation from Uni3D and robot pose, we generate action sequences through an iterative denoising process. We leverage DP3's existing conditioning mechanism to incorporate the visual features into the denoising network.

Feature-wise Linear Modulation (FiLM). Following DP3's architecture, visual representations are integrated into the diffusion process using Feature-wise Linear Modulation (FiLM)[8] conditioning. The concatenated features

(64-dimensional visual features $\mathbf{v}_t$ from Uni3D and 64-dimensional robot pose $\mathbf{q}_t$) are used to compute affine transformation parameters for each layer of the U-Net denoising network. Specifically, for each U-Net layer l, linear transformations produce scale γ_l and shift β_l parameters:

$$\mathbf{h}'_l = \gamma_l([\mathbf{v}_t; \mathbf{q}_t]) \odot \mathbf{h}_l + \beta_l([\mathbf{v}_t; \mathbf{q}_t]) \quad (1)$$

where $\mathbf{h}_l$ represents the intermediate features at layer l, $[\cdot;\cdot]$ denotes concatenation, and $\odot$ denotes element-wise multiplication. This conditioning mechanism allows the visual and proprioceptive context to modulate the denoising process at multiple scales throughout the U-Net architecture.

Diffusion Process. The U-Net-based noise prediction network ϵ_θ performs iterative denoising. Starting from a Gaussian noise sample, we perform multiple denoising steps:

$$\mathbf{a}_t^{k-1} = \alpha_k(\mathbf{a}_t^k - \gamma_k \epsilon_\theta(\mathbf{a}_t^k, k, \mathbf{v}_t, \mathbf{q}_t)) + \sigma_k \mathcal{N}(0, \mathbf{I}) \quad (2)$$

where k is the diffusion timestep, and α_k, γ_k, and σ_k are functions of k that define the noise schedule.

The training objective follows the standard diffusion model loss:

$$\mathcal{L} = \text{MSE}\left(\epsilon, \epsilon_\theta(\alpha_k \mathbf{a}_t^0 + \sigma_k \epsilon, k, \mathbf{v}_t, \mathbf{q}_t)\right) \quad (3)$$

where $\mathbf{a}_t^0$ is the ground truth action sequence from demonstrations, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is the noise sample.

During training, only the parameters θ of the U-Net noise prediction network and the linear projection layer are updated, while the Uni3D encoder remains frozen to preserve its pre-trained representations.

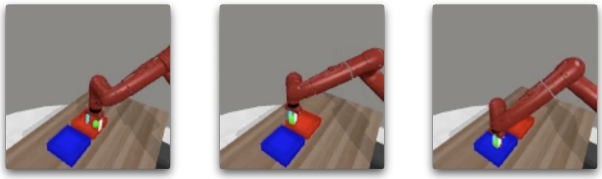
Implementation Details. We implement U3DP using the Uni3D-Small variant due to computational constraints, which provides an effective balance between model capacity and resource requirements. At inference time, we employ a receding horizon control strategy where we predict a sequence of 16 actions but execute only the first 8 steps before replanning, following DP3's protocol.

3.2. Baseline Method

We compare our approach with the original 3D Diffusion Policy (DP3) as our primary baseline. DP3 uses the same diffusion framework and training procedure as our method, with the key difference being the perception module. Instead of the pre-trained Uni3D model, DP3 employs a lightweight three-layer MLP encoder followed by max-pooling to process point clouds. This baseline allows us to isolate the impact of incorporating pre-trained 3D representations on policy learning performance.

4. Dataset and Features

MetaWorld [12] is a benchmark suite designed for meta-reinforcement learning and multi-task learning research. It provides a standardized collection of robotic manipulation tasks with consistent observation and action spaces, enabling fair comparison across different algorithms and approaches. These tasks are performed by a simulated Sawyer robotic arm within the MuJoCo physics engine. Among the benchmark tasks, we specifically target bin-picking task since the vanilla dp3 only achieves 50% success rate on these, and it would be a good experiment to see if we can achieve improvement on these task. Also, we aim to explore the generalizability of the visual foundation model for out-of-domain inference, and thistask can be easily modify for this purpose. For example, we can simply change the size or color of the cube for out-of-domain test.



(a) Block in red bin (b) Pick block (c) Put in blue bin

Figure 2. Metaworld Bin Picking Task

MetaWorld serves as an ideal benchmark for this project since the 3D Diffusion Policy framework already implements a MetaWorld wrapper as a tested benchmark, allowing us to focus on the novel aspects of our research rather than environment implementation details. Moreover, MetaWorld provides rich visual observations that align well with Uni3D's capabilities as a 3D foundation model.

5. Experiment and Analysis

5.1. Training Detail

To experiment how well the pretrain Uni3D 3d representation can be transfer to manipulation task, we try config with different axis:

- **Freeze:** Whether to freeze Uni3D during training
- **Load Pretrian weight:** If don't freeze Uni3D, whether to load pretrained checkpoint
- **Uni3D group size:** The size that Uni3D first group points into local patches with FPS (farthest point sampling) and kNN (k nearest neighbor), the original value is 32, we additionally experiment with 16 since the metaworld image have relatively low resolution (80*80)

We train U3DP with the configs in Table 1 and get the evaluation success rate as follows:

Here are some fix settings throughout the experiments:

- **Uni3D Model:** We use Uni3D-small with 23 millions parameters, because our GPU can't handle base model

Config	Model	Freeze	Pretrain	Group Size/Number
dp3	n/a	n/a	n/a	n/a
1	small	no	yes	16/128
2	small	yes	yes	16/128
3	small	no	no	16/128
4	small	no	yes	32/256

Table 1. Experiment configurations

Config	Epochs	Eval Demos	SR
dp3	3000	20	0.85
1	200	20	0.90
2	200	20	0.0
3	200	20	0.30
4	200	20	0.85

Table 2. Training Results

Uni3D Model	Small (23M)
Demos	100
Epochs	200
Optim	AdamW
Lr	1e-4
Weight Decay	1e-6
Eval	20
DP3 Hidden Dims	256/512/1024

Table 3. Training Details

- **Training Demos:** We generate 100 demos bin-picking with randomnized initial block position and use all of those during training.
- **Training Epoch:** Other than vanilla dp3 (3000 epochs), we train model with 200 epochs
- **DP3 Hidden Dims:** We half the size of the hidden dims from 512/1024/2048 to 256/512/1024 for faster training. We also confirm that halfing the size of the hidden dims won't harm vanilla dp3 performance.

From Table 2, we can observe that freezing the Uni3D model results in poor model performance with 0% success rate. However, fine-tuning Uni3D yields strong model performance that equals or even outperforms vanilla DP3 under identical training settings and demonstration data. This demonstrates that while the pretrained Uni3D representations contain valuable 3D geometric and semantic knowledge, direct transfer without fine-tune fails to capture the nuanced visual-motor relationships required for manipulation tasks.

The stark contrast between frozen and fine-tuned configurations suggests that the pretrained features, though rich in spatial understanding, require task-specific refinement to align with the action space and temporal dynamics of robotic manipulation. The success of the fine-tuned approach indicates that Uni3D's foundational representations provide a beneficial initialization that can be effectively adapted for policy learning, potentially offering improved sample efficiency compared to training from scratch. However, the failure of the frozen model highlights the domain gap between general 3D understanding tasks and the specific requirements of visual-motor control, suggesting that future work should focus on developing more sophisticated adaptation mechanisms that can better preserve and utilize pretrained knowledge while enabling effective policy learning.

5.2. Evaluation Test

To test the general ability of the U3DPs, we modify the original bin picking test with different settings to generate out of domain test environment.

- **Exp0:** Original environment setting (red $\rightarrow$ blue, right $\rightarrow$ left)
- **Exp1:** Exchange two bins color (blue $\rightarrow$ red, right $\rightarrow$ left)
- **Exp2:** Switch start and goal position, switch bin position (red $\rightarrow$ blue, left $\rightarrow$ right)
- **Exp3:** Switch start and goal position, do not switch bin (blue $\rightarrow$ red, left $\rightarrow$ right)
- **Exp4:** Move two bins further apart
- **Exp5:** Change green block to basketball
- **Exp6:** Change green block to red cylinder
- **Exp7:** Change green block to wooden block
- **Exp8:** Use two red bins
- **Exp9:** Use two blue bins
- **Exp10:** Add wooded block into bin (two blocks in bin)

The visualize experiment settings are shown in Figure 3.

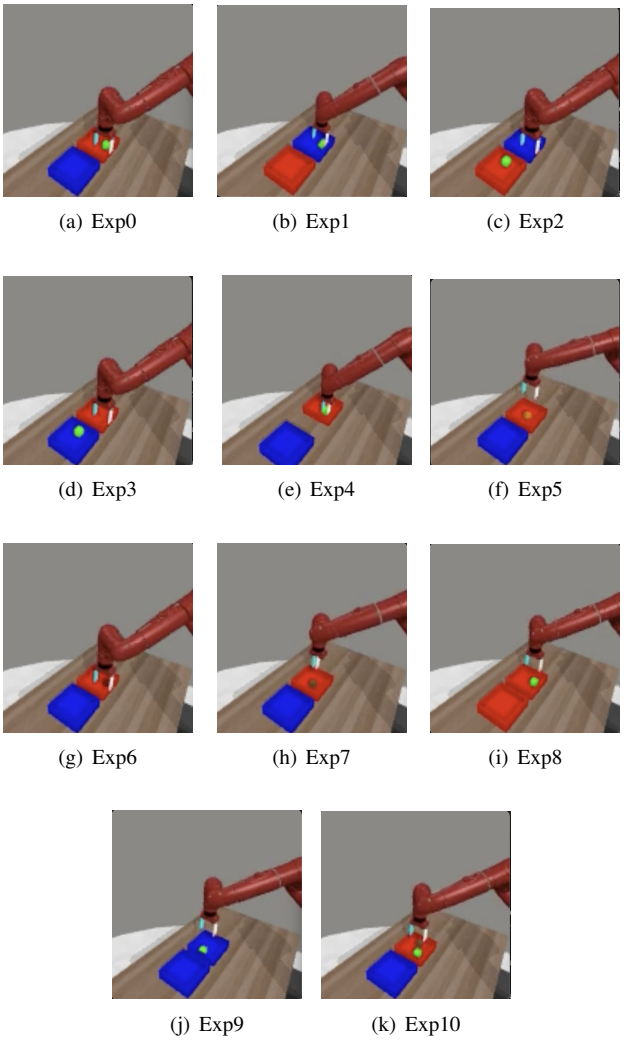
We can briefly divide these tests into three categories:

- **Color Variance:** (Exp 1, 7, 8, 9): Bins and blocks have no shape or position variance, but with different color or texture
- **Spatial Variance:** (Exp 2, 3, 4): The position of the bins are changed
- **Object Variance:** (Exp 5, 6, 10): The object being picked is changed

5.3. DP3 v.s. U3DP

Based on the experimental results shown in Figure 4, we can analyze the comparative performance of U3DP (Config 1) against vanilla DP3 across different generalization scenarios:

Color Variance Results: U3DP demonstrates superior robustness to color and texture variations compared to DP3. In Exp1 (color exchange), U3DP achieves 85% success rate versus DP3's 60%, indicating better adaptation to visual appearance changes. This advantage is particularly pronounced in Exp7 (wooden block) where U3DP achieves



(a) Exp0 (b) Exp1 (c) Exp2

(d) Exp3 (e) Exp4 (f) Exp5

(g) Exp6 (h) Exp7 (i) Exp8

(j) Exp9 (k) Exp10

Figure 3. All Bin Picking Task Experiment Setting

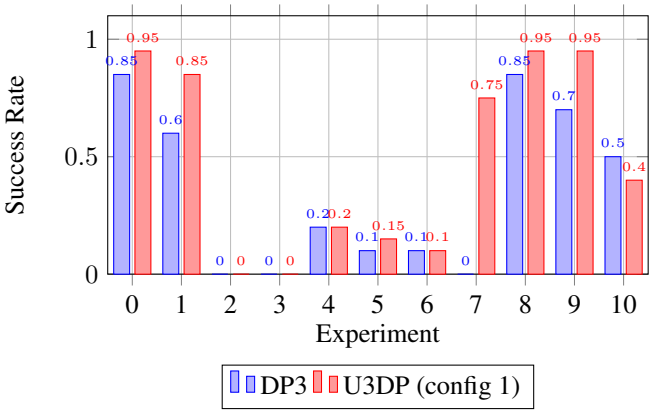


Figure 4. Comparison of experiment scores between `DP3` and `U3DP (config 1)`.

75% success while DP3 completely fails (0%), and in Exp9 (two blue bins) where U3DP maintains 95% success compared to DP3's 70%. Although vanilla DP3 can achieve similar generalizability if not using pointcloud color, we claim that color information could be extremely useful for other task like vision-language-action-model that required RGB dense feature.

Spatial Variance Results: Both models exhibit significant difficulties with spatial reconfiguration tasks. In Exp2 and Exp3, where start/goal positions are switched, both approaches achieve 0% success rate, indicating a fundamental limitation in spatial reasoning and adaptation. The modest 20% success rate for both models in Exp4 (increased bin separation) suggests that even minor spatial changes can severely impact performance, highlighting the brittle nature of current visual-motor policies to geometric variations. We argue that the spatial semantic representation within Uni3D is not transferred to robot manipulation task.

Object Variance Results: U3DP shows generally improved performance in object generalization scenarios. While both models struggle with basketball substitution (Exp5: U3DP 15% vs DP3 10%) and cylinder replacement (Exp6: both 10%), U3DP's superior performance with the wooden block (Exp7: 75% vs 0%) suggests that the 3D foundation model representations provide better semantic understanding for certain object categories.

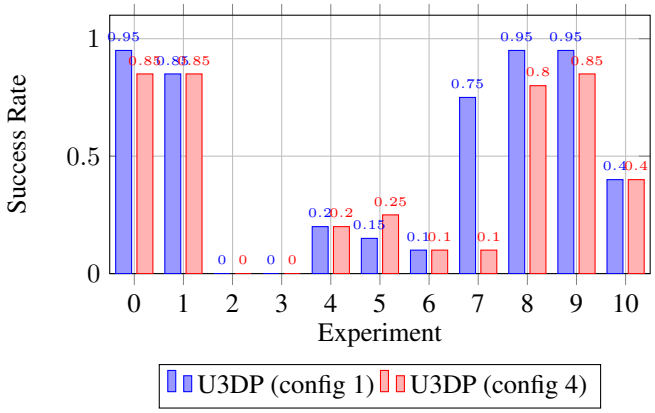


Figure 5. Comparison of experiment scores between `U3DP (config 1)` and `U3DP (config 4)`.

Based on the comparison between U3DP configurations with different Uni3D group sizes, we observe that Config 1 (group size 16) generally outperforms Config 4 (group size 32) across most experimental scenarios. This suggests that reducing the Uni3D group size from 32 to 16 is beneficial for the low-resolution MetaWorld environment (80×80 pixels). The smaller grouping appears to provide more appropriate local feature granularity that better matches the limited visual detail available in the simulation, preventing over-segmentation of sparse point cloud information. However, both configurations still exhibit identical failures in

spatial reconfiguration tasks (Exp2-3), indicating that group size optimization alone cannot address fundamental spatial reasoning limitations

5.4. Ablation Study: Bottom Adapter

Refer to [11], we test the influence of the bottom adapter which is a 6*6 fully connected network that process RGB pointcloud before sending into Uni3D.

Based on Table 4, we can observe the impact of the bottom adapter on model performance. Comparing Config 1 (without bottom adapter, 90% success rate) to Config ab2 (with bottom adapter, 40% success rate), both using identical settings except for the adapter, we find that the bottom adapter significantly degrades performance. This is probably because the additional layer enable the possibility to brake the well-normalized rgb and xyz value for Uni3D input.

On the other hand, when the Uni3D model is frozen, the bottom adapter provides some benefit (Config ab1: 30% vs Config 2: 0%), suggesting it can partially bridge the representation gap when no fine-tuning is allowed. However, this improvement is still substantially lower than the fine-tuned approach without the adapter.

The bottom adapter appears to introduce unnecessary complexity and potential information bottlenecks when Uni3D can be fine-tuned directly. The 6×6 fully connected network may constrain the rich RGB point cloud features before they reach the foundation model, limiting the model's ability to leverage the full representational capacity of Uni3D.

Config	Freeze	Pretrain	Group Size/Number	SR
dp3	n/a	n/a	n/a	0.85
1	no	yes	16/128	0.90
2	yes	yes	16/128	0.0
ab1	yes	yes	16/128	0.30
ab2	no	yes	16/128	0.40

Table 4. Experiment configurations

6. Conclusion and Future Work

Our experiments indicate that U3DP and DP3 achieve comparable overall success rates on the bin-picking benchmark, but U3DP exhibits noticeably greater robustness to variations in object and bin colors. This suggests that incorporating pre-trained 3D features can improve visual generalization; however, the limited scope of our current evaluation makes it difficult to draw definitive conclusions about how effectively Uni3D's pretraining transfers to robotic manipulation tasks.

With additional time, personnel, or computational resources, we would expand our study along three main axes. First, we would evaluate alternative point-cloud encoders to compare how varied pretraining objectives affect downstream policy performance. Second, we would explore multiple Uni3D model sizes (e.g., Small, Base, Large) to better understand the effectiveness of pretraining. Finally, we would experiment with diverse adapter configurations, including deeper adapter stacks and parameter-efficient fine-tuning methods such as LoRA [7], to identify adaptation strategies that preserve pre-trained knowledge while maximizing manipulation performance.

References

[1] A. Brock, S. De, S. L. Smith, and K. Simonyan. High-performance large-scale image recognition without normalization, 2021.

[2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners, 2021.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

[8] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[11] M. Sharma, C. Fantacci, Y. Zhou, S. Koppula, N. Heess, J. Scholz, and Y. Aytar. Lossless adaptation of pretrained vision models for robotic manipulation, 2023.

[12] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings*

of Machine Learning Research, pages 1094–1100. PMLR, 30 Oct–01 Nov 2020.

[13] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022.

[14] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang. Uni3d: Exploring unified 3d representation at scale, 2023.