Avi: A 3D Vision-Language Action Model Architecture generating Action from Volumetric Inference

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose **Avi**, a novel **3D Vision-Language-Action (VLA) architecture** that reframes robotic action generation as a problem of 3D perception and spatial reasoning, rather than low-level policy learning. While existing VLA models primarily operate on 2D visual inputs and are trained end-to-end on task-specific action policies, Avi leverages 3D point clouds and language-grounded scene understanding to compute actions through classical geometric transformations. This approach enables interpretable, generalizable behaviors that are robust to occlusions, camera pose variations, and changes in viewpoint. By treating the robotic decision-making process as a structured reasoning task over 3D representations, Avi bridges the gap between high-level language instructions and low-level actuation without requiring opaque policy learning. We evaluate Avi on a suite of 3D manipulation benchmarks and demonstrate its ability to generate semantically aligned, physically executable actions across diverse tasks and environments. Our results highlight the potential of 3D vision-language reasoning as a foundation for scalable, robust robotic systems. Check it out at action-volume-inference.github.io.

1 Introduction

Vision-Language-Action (VLA) models have recently gained significant attention in the robotics and machine learning communities Black et al., Team et al. [2025]. While these models have demonstrated impressive capabilities in connecting high-level natural language instructions with actionable robot policies, the vast majority of current VLAs operate solely on 2D image inputs. This reliance on 2D perception imposes fundamental limitations: reasoning about depth, object geometry, and fine-grained spatial relations becomes indirect and often error-prone.

In this work, we propose to move beyond the 2D paradigm by training a VLA model that *natively* operates on 3D representations, specifically point clouds. Our approach is built upon ShapeLLM, a 3D Multi-Modal Language Model (3D MMLM), which we finetune to condition on both natural language commands and a 3D point cloud of the scene. Rather than directly outputting low-level joint actions, the model predicts a *delta point cloud* that represents the desired post-condition of the manipulated object(s). Robot joint actions are then derived through inverse kinematics, aligning the end-effector to the predicted "after" state of the point cloud.

This design presents several key advantages compared to traditional 2D VLAs. First, it enables **finer-grained and more robust manipulation**: models without 3D priors often struggle with precise control or spatial reasoning (e.g., PI-0 failing to complete a pick-up action by stopping mid-air). Second, our method facilitates **more reliable sim-to-real transfer**. Training 2D VLAs typically requires large-scale real-world demonstrations through behavioral cloning, which is prohibitively expensive. By contrast, 3D point cloud representations are more invariant to appearance shifts such as lighting or texture changes, making them well-suited for sim-to-real transfer (a principle

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

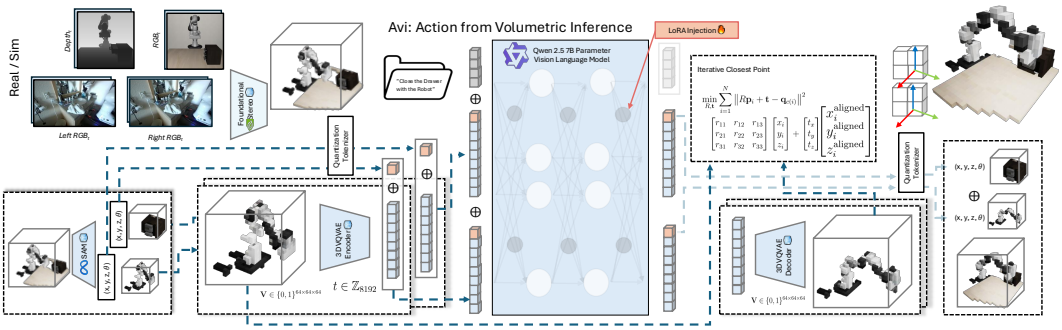

Figure 1: Overview of our Vision-Language Action Model **Avi**, a volumetric vision-language system for robotic action generation. Avi combines stereo reconstruction, 2D segmentation (via Segment Anything), and a fine-tuned 3D Vision-Language Model (based on Qwen-VL and 3D VQVAE embeddings) to predict goal-conditioned 3D volumes. We further align these volumes using classical geometric optimization (ICP) to produce interpretable, spatially grounded actions.

also emphasized in prior work, e.g., Dieter Fox's talks). Finally, our framework naturally supports **morphology-agnostic policies**. Since the model predicts geometric outcomes in the form of deformed point clouds rather than joint torques, the resulting actions can be computed via inverse kinematics across different robot embodiments.

In summary, we present two main contributions:

1. **AVI (Action from Volumetric Inference)**: a novel architecture that integrates a 3D Multi-Modal Language Model to infer actions through volumetric reasoning, rather than directly generating action tokens. This approach shifts the focus from language-to-action to language-to-geometry, enabling richer spatial grounding.
2. **Location Quantization for 3D MLLMs**: a general technique for discretizing spatial information that allows pretrained 3D MLLMs to generalize at the *object level* rather than at the scene level. Unlike current state-of-the-art methods, which often overfit to scene layouts, our quantization method enables modularity and reusability across different 3D architectures and robotics tasks.

2 Related Work

Recent work at the intersection of vision, language, and robotics has demonstrated remarkable progress, but our approach differs in important ways. Most prior Vision-Language-Action (VLA) methods directly predict robot-specific action tokens, trained to mimic collected demonstrations. For example, PerAct Shridhar et al. [2022], CLIPort Shridhar et al. [2021], and Transporter Networks Zeng et al. [2021] couple visual perception with learned policies tailored to specific robot morphologies. Similarly, Unified Video Action Black et al. and related works integrate video-based action representations, but remain dependent on large-scale, robot-specific data. This tight coupling creates a heavy reliance on mass data collection and labeling services (e.g., MaxInsights, akin to Scale AI). By contrast, our approach emphasizes a **morphology-agnostic policy**: rather than outputting actions, our model predicts transformed 3D point clouds from which robot-specific trajectories can be computed via inverse kinematics.

2D Pretraining. A significant line of research leverages 2D vision pretraining for robotics. Yen-Chen et al. Yen-Chen et al. [2020] demonstrated transfer learning in manipulation policies by reusing COCO-pretrained backbones. Radosavovic et al. Radosavovic et al. [2023] applied masked autoencoding to pretrain ViTs on large-scale 2D data before transferring to robotics. More recent work has focused on scaling such paradigms: Lift3D Jia et al. [2024] extends 2D representations into implicit depth and point-cloud embeddings, while 3D-MVP Qian et al. [2024] and FVP Hou et al. [2025] extend multiview pretraining to robotic manipulation and 4D video, respectively. World-consistent diffusion models Zhang et al. [2025] also highlight the utility of 2D/temporal video

modeling as a precursor to structured 3D reasoning. While effective, these approaches remain fundamentally limited by their reliance on 2D visual input for geometry.

3D Pretraining. The shift toward native 3D pretraining has led to a surge of new models. Recon++ Qi et al. [2023] pioneered contrastive representation learning for point clouds. ShapeLLM Qi et al. [2024] builds on Recon++ with ChatGPT-4V generated prompts and LLaMA backbones, surpassing PointLLM Xu et al. [2024]. JEPA Saito et al. [2025] introduces predictive joint embedding architectures, while SUGAR Chen et al. [2024] pretrains a transformer encoder from scratch on a massive dataset of 752.2K single objects and 110.7K multi-object scenes. Other works explore integrating LMMs with 3D input, such as LLaVA-3D Zhu et al. [2024], VoxPoser Huang et al. [2023], and PointVLA Li et al. [2025], which directly inject 3D priors into vision-language models. These methods highlight the growing consensus that 3D pretraining provides stronger grounding for manipulation than 2D alone.

3D RL and VLA Models. At the intersection of 3D perception and policy learning, FP3 Yang et al. [2025] and DP3 focus on point-cloud-conditioned diffusion policies, pretrained on large robot datasets such as Droid. 3D-VLA Zhen et al. [2024] and SpatialVLA Qu et al. [2025] extend this direction by predicting volumetric or depth-infused representations for language-conditioned action. Recent embodied generalist agents Huang et al. [2024] and Gemini Robotics Team et al. [2025] scale VLA models across tasks, but remain heavily compute- and data-intensive. Meanwhile, industrial efforts such as Google Robotics and Nvidia GR00T are developing proprietary foundational VLA systems at scale. These approaches showcase the trend toward 3D-aware VLA, but most remain tied to *action token prediction*, which restricts generality.

In summary, prior work spans 2D pretraining, 3D representation learning, and 3D RL/VLA models. Our method diverges by reframing VLA as **language-to-geometry**: predicting 3D volumetric transformations instead of action tokens. This shift enables morphology-agnostic control, efficient sim-to-real transfer, and more robust grounding in spatial reasoning.

3 Method

Our architecture (Figure 1) extends the foundational 3D model, **ShapeLLM-Omni**. ShapeLLM-Omni is pretrained on large-scale 3D assets and is capable of handling multi-modal inputs, including text, images, and most importantly, 3D point clouds. In its original formulation, ShapeLLM-Omni was primarily trained on *single-object* 3D assets. By contrast, robotic environments typically consist of *multi-object* scenes with complex spatial relationships, requiring extensions for effective reasoning and manipulation.

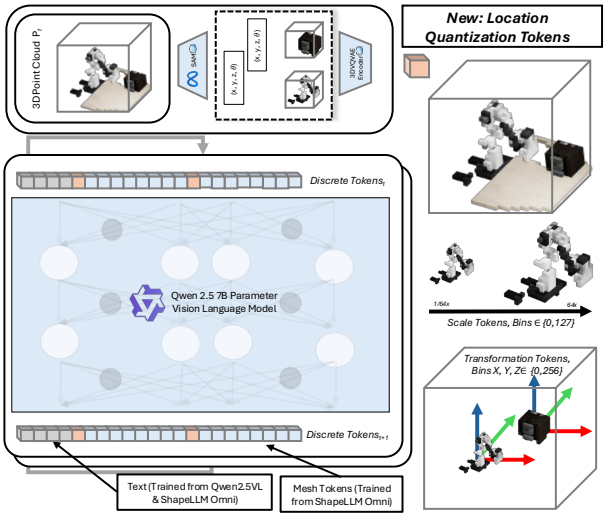

Figure 2: The 3D MLLM training stage with the new extended vocabulary is detailed, along with the discrete tokens represented between t and $t+1$. The new Location Quanization tokens are detailed, including the Transformation and Scale Tokens.

Formally, we represent each scene as a point cloud

$$\mathcal{P} \subset \mathbb{R}^{N\times 3},$$

where N denotes the number of points sampled in the scene. For simulation and data collection, we build on top of `Robosuite`, which provides realistic physics-based robotics tasks.

Our model processes both geometric and linguistic inputs by mapping them into a shared latent space $\mathcal{Z}$. Let $\mathcal{P}$ denote the input point

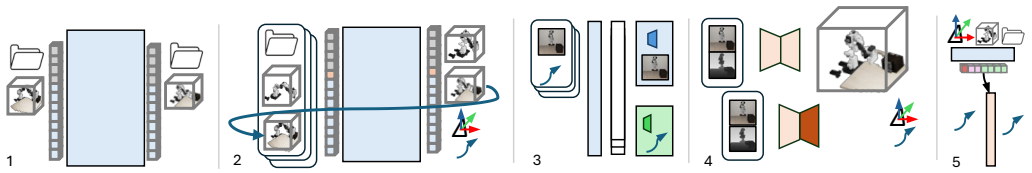

Figure 3: Comparison against related work. (1) describes Shape LLM Omni. (2) describes our work, Avi. (3) describes the Unified Video Action Model. (4) describes Robot 4D Generation. (5) describes 3D Foundation Policy, FP3.

cloud and $\mathcal{T}$ the human-provided text prompt. We define modality-specific encoders such that:

$$f_{\text{3D}}(\mathcal{P}) \in \mathcal{Z}, \quad f_{\text{text}}(\mathcal{T}) \in \mathcal{Z},$$

where f_{3D} and f_{text} are encoders for the 3D point cloud and text, respectively. This joint embedding ensures that both geometry and language are represented in a unified feature space, enabling cross-modal reasoning.

For the language backbone, we integrate **Qwen-2.5 (7B)**, a state-of-the-art large vision-language model with 7 billion parameters. Qwen-2.5 provides strong multi-turn instruction-following, chain-of-thought reasoning, and multilingual capabilities, making it particularly well-suited for language-conditioned robotics. By coupling Qwen-2.5 with ShapeLLM-Omni, our architecture inherits both powerful linguistic grounding and native 3D spatial reasoning, which are critical for precise manipulation in multi-object robotic environments.

We freeze the pretrained VQ-VAE encoder, where VQ-VAE stands for *Vector Quantized Variational Autoencoder*. This encoder maps a voxelized 3D shape $\mathcal{V} \in \mathbb{R}^{64\times64\times64}$ into a discrete latent representation consisting of 8192 tokens:

$$\text{Encoder}_{\text{VQ-VAE}}(\mathcal{V}) \to \mathbf{z} = [z_1, z_2, \ldots, z_{8192}], \quad z_i \in \mathcal{C}$$

where $\mathcal{C}$ is a learned codebook of latent embeddings.

The token sequence $\mathbf{z}$ is then passed through the VQ-VAE decoder to reconstruct the voxel grid $\hat{\mathcal{V}}$, which is subsequently converted back into a point cloud $\hat{\mathcal{P}} \subset \mathbb{R}^{N\times3}$.

3.1 Location Quantization

We maintain the initial token embeddings from the previously trained 3D Multi-Modal Large Language Model (3D-MLLM) *ShapeLLM-Omni*, ensuring compatibility with the pretrained architecture. To incorporate additional spatial and geometric information, we extend the vocabulary by introducing dedicated *position* and *scale* tokens.

Specifically, we define three independent position axes: $X, Y, Z \in \{1, 2, \ldots, 256\}$, each discretized into 256 bins. This introduces a total of 768 new tokens corresponding to positional context. In addition, we discretize the object scale into $S \in \{1, 2, \ldots, 128\}$, yielding 128 scale tokens. Thus, the overall vocabulary extension equates 896 additional tokens.

Finally, the extended embedding matrix becomes $E' \in \mathbb{R}^{(|\mathcal{V}_0|+896)\times d}$, where $|\mathcal{V}_0|$ denotes the size of the original vocabulary from ShapeLLM-Omni and d is the embedding dimension. The new embeddings corresponding to the 896 tokens are initialized (e.g., randomly or via scaled normal initialization), while the pretrained embeddings for the original vocabulary are preserved to retain the knowledge of the base model.

Figure 2 illustrates the *Location Quantization* method that is incorporated into both the training and inference stages of our framework. In this approach, each object within the scene is first identified and segmented using the `Segment Anything` model (SAM), which provides fine-grained object boundaries. For every segmented object, we attach a set of quantized location tokens that encode its spatial context within the 3D environment. These tokens serve as an additional input modality, enabling the model to reason not only about the object's semantic identity but also about its discretized position and scale relative to the overall scene. This mechanism ensures that spatial information is consistently represented throughout the training and evaluation process, thereby enhancing the model's capacity for grounding and generalization in downstream tasks.

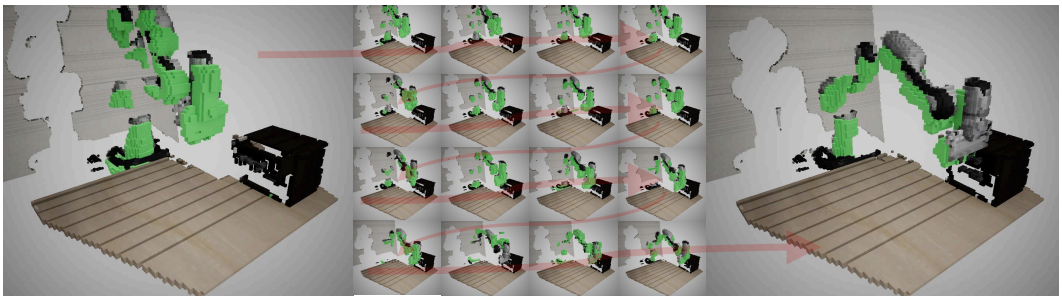

Figure 4: The **Left Image** represents the starting position of the scene. The green voxels represent the predicted next time stamp. The **Right Image** represents the end time stamp, and the series of images in between indicate the rollout.

3.2 Transformation Calculation

Given a prompt and current scene point cloud $\mathcal{P}_t$, we generate a next point cloud prediction $\hat{\mathcal{P}}_{t+1}$ such that: $\hat{\mathcal{P}}_{t+1} \approx \mathcal{P}_t + \Delta\mathcal{P}$ where $\Delta\mathcal{P}$ represents the learned spatial change conditioned on the prompt. We then compute the Iterative Closest Point (ICP) transformation, defined as the rigid transformation (R, t) that minimizes the alignment error:

$$\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\|^2$$

where $\{\mathbf{x}_i\}$ are points from the source point cloud and $\{\mathbf{y}_i\}$ are their closest points in the target point cloud, $\mathbf{R} \in SO(3)$ is a rotation matrix, and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. The resulting transformation is then applied to the robot's end effector position (X, Y, Z), and the updated pose is executed.

4 Setup and Experimentation

We fine-tune the foundational model on robotics training data using a single NVIDIA A6000 GPU with 48GB of memory, which is sufficient to accommodate the pretrained **ShapeLLM-Omni** backbone. For training, we utilize the **LIBERO Dataset** Liu et al. [2023], which provides diverse task demonstrations within the `Robosuite` environment. Each demonstration contains synchronized RGB-D observations and robot proprioceptive states for a Franka Panda manipulator. In our experiments, we select 50 demonstrations corresponding to the drawer-closing task. Figure 4 illustrates eighteen separate inference rollouts for this task, where the robot is required to align its end-effector and successfully close the drawer.

To approximate real-world deployment, the process of generating 3D voxelized representations can be reproduced using **FoundationStereo**, which recovers high-fidelity 3D structure from stereo RGB inputs. This ensures that our approach is not limited to simulation, but can extend naturally to real-robot scenarios.

For segmentation, we employ the **Segment Anything Model (SAM)** to isolate individual objects from raw visual inputs. To maintain stability and avoid overfitting, the SAM encoder weights are frozen throughout training, providing consistent object-level features while the rest of the model adapts to robotics-specific tasks.

To regularize training in this limited-data regime, we apply dropout with a probability of $p = 0.05$. Fine-tuning is performed using **Low-Rank Adaptation (LoRA)** to enable efficient parameter updates without full model retraining. Specifically, we unfreeze the last K layers of the attention mechanism, inserting LoRA adaptation matrices into the Query (Q), Key (K), and Value (V) projection layers. This design allows the model to adapt effectively to manipulation while retaining the broad multimodal reasoning capabilities of the pretrained 3D MLLM.

Table 1: Comparison of our approach with related methods in vision-language or 3D robotic policy learning.

Method	Input Modality	Core Mechanism	Generates 3D Point Clouds?	No Action Tokens Needed?
This Work (Avi)	3D point clouds + language	3D MLLM predicting delta point clouds + IK	**Yes**	**Yes**
Robot4DGen	RGB-D video	4D video generation with multi-view constraint	No (video only)	No
Unified Video-Action (UVA)	RGB video	Joint video–action latent modeling	No	No
DP3	3D point clouds	Diffusion model over actions conditioned on 3D	Uses 3D conditioning, outputs actions	No
FP3	3D point clouds + language	Diffusion transformer policy pre-trained on 3D	No (actions directly)	No

5 Results

We present preliminary qualitative results of our proposed architecture. Figure 4 illustrates the rollout of the drawer-closing task across eighteen inference steps. The leftmost panel depicts the initial state of the scene, while the rightmost panel shows the final state after execution. Intermediate frames visualize the predicted voxelized "delta" states (shown in green), which are progressively aligned with the ground-truth trajectory. These results demonstrate that Avi is able to generate semantically consistent and physically realizable action trajectories conditioned on natural language instructions.

In addition, we evaluate the effect of the *location quantization* mechanism through qualitative ablations (Section 6). The comparisons reveal that quantized spatial tokens significantly improve fine-grained manipulation accuracy, particularly in precision tasks such as grasping and drawer closing. Without explicit location tokens, the model often produces ambiguous intermediate states, leading to suboptimal or incomplete executions. With quantization enabled, the model more reliably grounds spatial relationships, resulting in consistent end-effector alignment and successful task completion.

6 Qualitative Ablation Studies

We conduct a qualitative ablation study to evaluate the necessity of the proposed *location quantization* mechanism. Figure 5 presents a side-by-side comparison: the left panel illustrates our full architecture with location quantization enabled, while the right panel shows the same model architecture without this component.

The results demonstrate that location quantization is critical for precise manipulation. In tasks requiring fine-grained control, such as pick-and-place or insertion, the model must accurately predict gripper positions in order for gripper open and close states to be correctly inferred from visual input alone. Without explicit location tokens, the model struggles to resolve subtle spatial relations, often leading to ambiguous predictions that fail to translate into correct end-effector motions. By contrast, with location quantization, the model leverages discretized spatial embeddings to ground geometric reasoning, resulting in more reliable alignment of the gripper with the target object and more consistent execution of precision tasks.

7 Conclusion

In this work, we introduced **Avi**, a novel 3D Vision-Language-Action (VLA) architecture that reframes robotic control as a problem of volumetric reasoning rather than low-level policy generation. By leveraging ShapeLLM-Omni as a 3D Multi-Modal Language Model and extending it with location quantization, we enable the model to interpret natural language instructions and predict

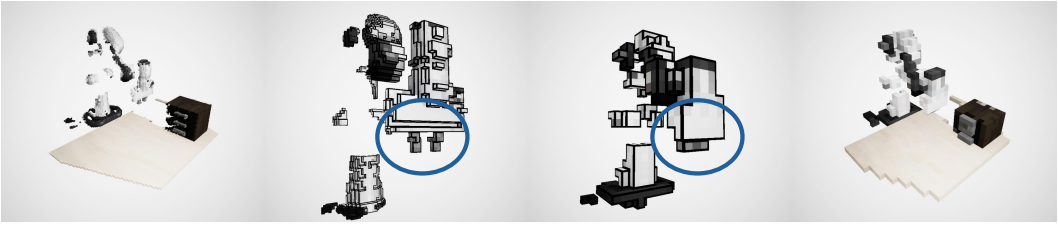

Figure 5: The **Left Images** contain the architecture with location quantization, including both the world scene and the robot. The **Right Images** include the architecture without location quantization.

goal-conditioned 3D representations of the environment. These predicted volumes are then aligned through geometric optimization, yielding interpretable and morphology-agnostic actions.

Our experiments on the LIBERO benchmark demonstrate that Avi produces semantically consistent and physically realizable manipulations from only a small number of demonstrations. Through ablation, we further show that the proposed location quantization mechanism is critical for grounding spatial relations and achieving robust performance in precision tasks.

Overall, this work highlights the potential of 3D vision-language reasoning as a foundation for scalable, generalizable, and robust robotic systems. In future work, we plan to extend Avi to multi-task and multi-robot settings, evaluate it under real-world deployment using stereo-based 3D reconstruction pipelines, and integrate reinforcement learning to further refine long-horizon planning capabilities.

References

K Black, N Brown, D Driess, A Esmail, M Equi, C Finn, N Fusai, L Groom, K Hausman, B Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, oct. 2024. *URL http://arxiv. org/abs/2410.24164*.

Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Sugar: Pre-training 3d visual representations for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18049–18060, 2024.

Chengkai Hou, Yanjie Ze, Yankai Fu, Zeyu Gao, Yue Yu, Songbo Hu, Shanghang Zhang, and Huazhe Xu. Fvp: 4d visual pre-training for robot learning. *ICCV*, 2025.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024.

Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *arXiv preprint arXiv:2503.07511*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS 2023 — Datasets and Benchmarks Track*, 2023.

Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning (ICML)*, 2023.

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. *arXiv preprint arXiv:2402.17766*, 2024.

Shengyi Qian, Kaichun Mo, Valts Blukis, David F Fouhey, Dieter Fox, and Ankit Goyal. 3d-mvp: 3d multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2406.18158*, 2024.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

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

Ayumu Saito, Prachi Kudeshia, and Jiju Poovvancheri. Point-jepa: Joint embedding predictive architecture for 3d point cloud self-supervised learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

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

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *ECCV*, 2024.

Rujia Yang, Geng Chen, Chuan Wen, and Yang Gao. Fp3: A 3d foundation policy for robotic manipulation, 2025. URL `https://arxiv.org/abs/2503.08950`.

Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7293. IEEE, 2020.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

Qihang Zhang, Shuangfei Zhai, Miguel Angel Bautista Martin, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3d modeling. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21685–21695, 2025.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: **[TODO]**

   Justification: **[TODO]**

   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: **[TODO]**

   Justification: **[TODO]**

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: **[TODO]**

   Justification: **[TODO]**

   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: **[TODO]**

   Justification: **[TODO]**

   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: **[TODO]**

   Justification: **[TODO]**

   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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   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: **[TODO]**

   Justification: **[TODO]**

   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: **[TODO]**

   Justification: **[TODO]**

   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: **[TODO]**

   Justification: **[TODO]**

   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: **[TODO]**

    Justification: **[TODO]**

    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: **[TODO]**

    Justification: **[TODO]**

    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: **[TODO]**

    Justification: **[TODO]**

    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: **[TODO]**

    Justification: **[TODO]**

    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: **[TODO]**

    Justification: **[TODO]**

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: **[TODO]**

    Justification: **[TODO]**

    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 rigorousness, or originality of the research, declaration is not required.

    Answer: **[TODO]**

    Justification: **[TODO]**

    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.