

GLaD: Geometric Latent Distillation for Vision-Language-Action Models

Minghao Guo, Meng Cao, Jiachen Tao, Rongtao Xu, Yan Yan, Xiaodan Liang, Ivan Laptev, Xiaoju Chang

arXiv:2512.09619v1 [cs.RO] 10 Dec 2025

Abstract—Most existing Vision-Language-Action (VLA) models rely primarily on RGB information, while ignoring geometric cues crucial for spatial reasoning and manipulation. In this work, we introduce GLaD, a geometry-aware VLA framework that incorporates 3D geometric priors during pretraining through knowledge distillation. Rather than distilling geometric features solely into the vision encoder, we align the LLM’s hidden states corresponding to visual tokens with features from a frozen geometry-aware vision transformer (VGVT), ensuring that geometric understanding is deeply integrated into the multimodal representations that drive action prediction. Pretrained on the Bridge dataset with this geometry distillation mechanism, GLaD achieves 94.1% average success rate across four LIBERO task suites, outperforming UniVLA (92.5%) which uses identical pretraining data. These results validate that geometry-aware pretraining enhances spatial reasoning and policy generalization without requiring explicit depth sensors or 3D annotations.

Index Terms—Vision-Language-Action Models, Pretraining, Geometry Distillation, Robot Manipulation, Spatial Reasoning.

I. INTRODUCTION

VISION-LANGUAGE-ACTION (VLA) models have emerged as a promising paradigm for embodied intelligence, enabling robots to generate control actions directly from visual observations and natural language instructions. Recent works [1]–[4] have demonstrated impressive performance on diverse manipulation tasks by leveraging large-scale multimodal pretraining. These models typically combine powerful vision encoders [5]–[7] and large language models to learn generalizable visuomotor policies from extensive robot demonstration datasets.

Despite these advances, current VLA architectures fundamentally lack *geometric understanding*, which represent the capability of perceiving spatial positions, 3D structures, and relational arrangements among objects in a scene—knowledge that is essential for robots to reason about where objects are, how they relate to each other, and how to interact with them effectively. Most VLAs rely on vision encoders pretrained with 2D contrastive objectives such as CLIP [5] or SigLIP [7], which excel at capturing semantic correspondences between images and text but do not encode 3D spatial information. These 2D embeddings represent visual scenes as flat semantic patterns without explicitly modeling depth, object poses, or

Minghao Guo, Rongtao Xu, Xiaodan Liang, Ivan Laptev and Xiaoju Chang are with MBZUAI, Abu Dhabi, United Arabic Emirates (email: minghao.guo, rongtao.xu, xiaodan.liang, ivan.laptev, xiaoju.chang@mbzui.ac.ae).

Meng Cao is with MBZUAI, Abu Dhabi, United Arabic Emirates (email: mengcaopku@gmail.com).

Jiachen Tao and Yan Yan are with University of Illinois Chicago, Chicago, United States (email: jtao26, yyan55@uic.edu).

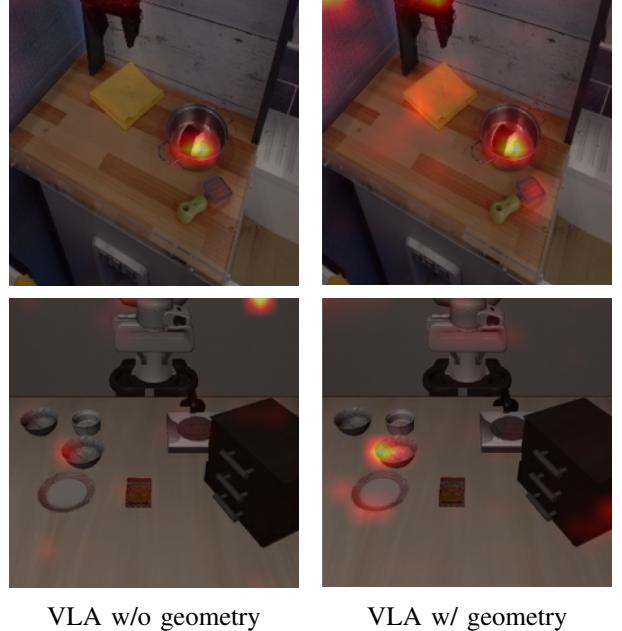


Fig. 1: Attention maps of VLA. Up: (Bridge scene) Move the table cloth from corner to edge of the table. Down: (LIBERO scene) Pick up the black bowl between the plate and the ramekin and place it on the plate.

spatial relationships—information that is critical for manipulation tasks where precise positioning matters, thus resulting wrong attention of the objects in the scene shown in Fig. 1. This raises a critical question: *Can we inject geometric priors into VLA pretraining to enhance scene understanding and improve policy generalization?*

To address this challenge, we propose **GLaD**, Geometric Latent Distillation vision-language-action framework that incorporates 3D geometric knowledge. Our key insight is that integrating geometric priors through knowledge distillation can substantially enhance a VLA’s ability to understand scene structure and reason about manipulation tasks. Specifically, GLaD introduces a geometry distillation mechanism during pretraining: we employ frozen VGVT [8], a pretrained model that directly infers 3D geometric attributes including depth maps, point clouds, and camera parameters from visual observations, as a teacher network to guide the learning of geometry-aware features. Critically, rather than adding geometric knowledge along with DINO-SigLIP features into LLM, we align the *LLM’s hidden states* corresponding to visual tokens with VGVT’s geometric features. This design ensures that geometric understanding is deeply integrated into the

multimodal representations that drive action prediction, rather than remaining isolated in the visual processing pipeline. The model is trained with a combined objective that simultaneously optimizes latent action prediction and geometry alignment, enabling it to learn both task-specific visuomotor skills and generalizable geometric reasoning.

We conduct extensive experiments on LIBERO [9] and LIBERO-PRO [10]. On LIBERO, a standard benchmark for language-conditioned manipulation across four task suites, GLaD achieves **94.1%** average success rate, outperforming UniVLA (92.5%) which uses both identical pretraining and posttraining data, and substantially surpassing other strong baselines including OpenVLA (76.5%), Octo (75.1%), and Diffusion Policy (72.4%). Notably, GLaD demonstrates particularly strong performance on object manipulation tasks, achieving **97.4%** success rate on LIBERO-OBJECT, the highest among all evaluated methods. On LIBERO-PRO, a robustness benchmark that introduces controlled perturbations across object appearance, spatial layout, language semantics, and task composition to distinguish genuine task understanding from mere memorization, GLaD exhibits substantially improved robustness to visual appearance variations. Under object perturbations that modify color, texture, and size while preserving semantic equivalence, GLaD achieves **81%** success rate on LIBERO-GOAL compared to UniVLA’s 62%, and **54%** on LIBERO-LONG versus 47%—with specific tasks showing up to 60 percentage point improvements. These results validate that geometry-aware pretraining enables the model to learn intrinsic geometric features and manipulation affordances rather than relying on superficial visual characteristics, enhancing policy generalization beyond pattern matching.

Our main contributions are as follows:

- We identify a critical limitation in current VLA architectures, *i.e.*, the lack of geometric understanding due to reliance on 2D vision encoders (*e.g.*, CLIP, SigLIP) that do not encode spatial positions and object relations. We demonstrate that injecting geometric priors during pretraining can substantially enhance scene understanding and policy generalization.
- A geometry-aware VLA framework GLaD is proposed to incorporate 3D geometric knowledge through knowledge distillation. By leveraging VGGT as a frozen teacher network, we distill geometric features into the LLM’s hidden states corresponding to visual tokens, ensuring geometric understanding is deeply fused into the multimodal representations that drive action prediction, without requiring depth sensors or explicit 3D annotations.
- GLaD achieves an average success rate of 94.1% on the LIBERO benchmark, surpassing the baseline model UniVLA (92.5%). Furthermore, on the LIBERO-PRO robustness benchmark, GLaD demonstrates substantially improved resilience to visual appearance variations, achieving 81% on LIBERO-GOAL under object perturbations (vs 62% for UniVLA), validating the generalization capability of the proposed geometry-aware pretraining.

II. RELATED WORKS

Vision-Language-Action Models. Recent studies have extended large vision-language models (VLMs) to build general-purpose robotic policies capable of generating actions directly from visual and textual inputs. Early works [1]–[4], [11]–[18] primarily learn from 2D visual observations, relying on implicit reasoning over spatial structures. Subsequent research began to explicitly incorporate 3D spatial information to enhance spatial understanding and cross-embodiment generalization. For instance, SpatialVLA leverages RGB-D inputs and a Depth API [19], OG-VLA transforms multi-view RGB-D observations into point clouds and orthographic projections [20], PointVLA directly consumes point cloud data [21], and 4D-VLA extends this idea by integrating temporal sequences of RGB-D inputs [22]. More recent works attempt to implicitly encode 3D geometry without requiring explicit depth sensors: SpatialBot estimates depth using ZoeDepth [23], [24], 3D-VLA learns to infer spatial representations internally [25], and GeoVLA reconstructs 3D embeddings from 2D images through depth estimation and point-cloud generation [26]. Despite these advances, achieving consistent alignment between 3D spatial representations, 2D visual features, and language instructions remains a fundamental challenge in developing unified and robust VLA frameworks.

Geometry-Aware Visual Representation Learning. A large body of works focus on learning 3D geometry from 2D images. Representative tasks include monocular depth estimation [24], [27], [28], normal prediction [29]–[32], and single-view or multi-view reconstruction with implicit 3D representations such as NeRF [33] and neural surface fields. These approaches demonstrate that rich geometric priors can be extracted directly from RGB inputs. Building upon these foundations, recent geometry-grounded vision models [8], [34]–[38] aim to learn latent features that explicitly encode 3D scene structure. Notably, VGGT [8] jointly predicts depth, point clouds, and camera parameters from image sequences, producing geometry-aware representations with strong spatial consistency, while PI3 [34] learns permutation-invariant geometric embeddings. Such models provide powerful and generalizable geometric priors, making them suitable teachers for distilling 3D structure into downstream representation learning. However, effectively integrating these geometric priors into vision-language-action models without compromising their generalization capability remains an open challenge.

Knowledge Distillation. Before the advent of large language models, knowledge distillation was primarily used as a model compression technique, transferring soft predictions [39], intermediate features [40], [41], or relational structures [42] from a large teacher to a smaller student for efficient deployment. In the LLM era, distillation has expanded from compressing architectures to transferring capabilities, enabling smaller or specialized models to inherit instruction-following, reasoning, and alignment behaviours from powerful foundation models. Existing approaches can be grouped into three broad families: (1) generation-based supervision [43], where teachers provide large-scale labeled or synthesized instruction-response data through labeling [44]–[50], expansion [51]–[57], or

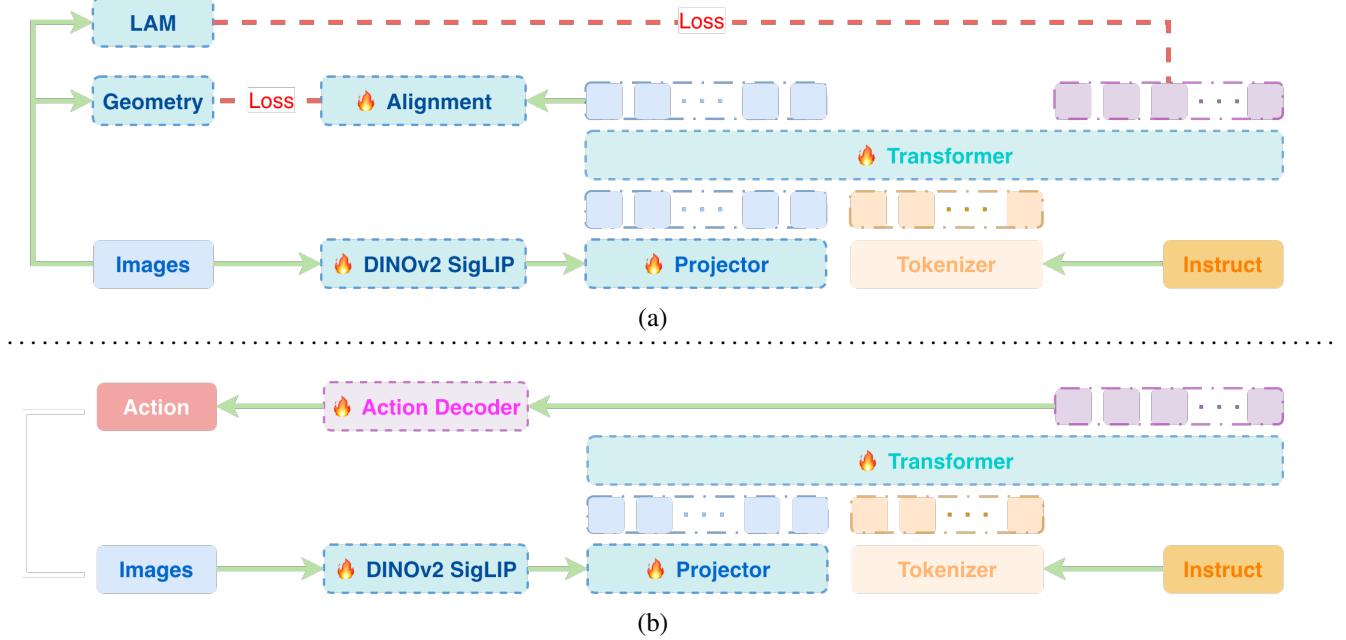


Fig. 2: GLaD model architecture. (a) **Pretraining stage:** The vision encoder (DINOv2 [6] + SigLIP [7]), projector, and LLM backbone (LLaMA-2-7B [85]) are trained, while the frozen VGGT teacher provides 3D geometric supervision. The feature alignment module learns to align LLM hidden states with VGGT features. (b) **Posttraining stage:** The VLA backbone is adapted via LoRA, while the action decoder and feature alignment module are fully trained. The VGGT remains frozen to preserve geometric priors.

curated generation [58]–[65]; (2) representation-level alignment, which aligns hidden states, output distributions, or preference signals through feature-based [43], [66]–[68] or feedback-based objectives [55], [67], [69]–[75]; and (3) self-bootstrapped distillation, where models iteratively refine their own generations without a stronger teacher [76]–[81]. Extending these trends to the vision-language-action domain, recent work investigates how to distill geometric priors from pretrained 3D models into multimodal models to overcome the inherent 2D bias of their visual encoders. Spatial Forcing aligns intermediate VLA representations with geometric embeddings to improve spatial precision in robotic control [82]; Vid-LLM injects reconstruction-derived geometric cues into video-based multimodal LLMs for enhanced 3D scene reasoning [83]; and 3D-Aware VLMs with Geometric Distillation transfer sparse correspondences, depth relations, and cost volumes into vision–language models to augment their 3D spatial understanding [84].

III. METHODOLOGY

We propose GLaD, an end-to-end VLA framework that integrates an LLM backbone, a vision encoder, an action head, and a geometry distillation module to enable the LLM to extract geometric information from images and generate latent actions conditioned on embodied task instructions. In Section III-A, we present our geometry distillation module that enhances the VLA with 3D geometric understanding. In Section III-B, we detail the training strategy.

A. Geometry Distillation

The overall architecture of GLaD is illustrated in Fig. 2. Our VLA backbone follows the UniVLA architecture [11], comprising a Prismatic vision encoder (DINOv2 [6] and SigLIP [7]), an MLP projector, LLaMA-2-7B backbone [85], and an action decoder. To enhance this backbone with 3D geometric understanding, we introduce a geometry distillation module that aligns the LLM’s internal visual representations with features from a pretrained geometry-aware teacher network. This module comprises two subcomponents:

1) *VGGT Feature Extractor:* Following 3DRS, we adopt pretrained VGGT as the teacher network for 3D geometry representation. Given a sequence of historical frames $\{o_{t-T}, \dots, o_t\}$ ($T = 32$), VGGT produces a spatio-temporal representation $\mathbf{F}_{3d} \in \mathbb{R}^{T \times L \times d_{vggt}}$ with $d_{vggt} = 2048$. In GLaD, only a single historical frame is used for simplicity. The VGGT parameters remain frozen throughout the training process.

For the VGGT temporal features \mathbf{F}_{3d} , we first apply adaptive pooling to match the spatial dimension with N_p (i.e., the number of visual patches). A “last-frame” aggregation strategy is then applied to generate a single-frame representation $\mathbf{F}_{3d}^{\text{single}} \in \mathbb{R}^{N_p \times d_{vggt}}$.

2) *Feature Alignment Network:* This network projects the final-layer LLM hidden states corresponding to image tokens into the VGGT feature space via a two-layer MLP:

$$\mathbf{H}_{\text{aligned}} = \text{MLP}(\mathbf{H}_{\text{img}}) \in \mathbb{R}^{N_p \times d_{vggt}}, \quad (1)$$

where $\mathbf{H}_{\text{img}} \in \mathbb{R}^{N_p \times d_{llm}}$ denotes the LLM hidden states at image token positions. We extract features from LLM hidden

states rather than the vision encoder to ensure geometric knowledge is integrated into fused multimodal representations.

3) *Training Objective*: During pretraining, GLaD optimizes a combined loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{VLA}} + \lambda \mathcal{L}_{\text{distill}}, \quad (2)$$

where \mathcal{L}_{VLA} is the cross-entropy loss for latent action prediction:

$$\mathcal{L}_{\text{VLA}} = - \sum_{i=1}^N \log P(\alpha_i | o, l, \alpha_{<i}), \quad (3)$$

and $\mathcal{L}_{\text{distill}}$ is the MSE loss for geometry alignment:

$$\mathcal{L}_{\text{distill}} = \|\mathbf{H}_{\text{aligned}} - \mathbf{F}_{3d}^{\text{single}}\|_2^2. \quad (4)$$

The hyperparameter λ balances action prediction and geometric alignment. The VGGT teacher remains frozen throughout training.

B. Training Strategy

Our training procedure consists of two stages: large-scale pretraining with geometry distillation and task-specific post-training.

1) *Stage 1: Pretraining with Geometry Distillation*: During the pretraining stage (Fig. 2(a)), GLaD is trained on a large-scale robotic manipulation dataset with the combined loss function described in Section III-A3. The model learns to predict latent actions while simultaneously aligning its internal visual representations with VGGT geometry features.

We initialize the VLA backbone with the pretrained Uni-VLA checkpoint and introduce the learnable alignment network. The VGGT teacher network remains frozen throughout pretraining. Training is conducted for 45 epochs using AdamW optimizer with a learning rate of 5e-7 on 8×A100 GPUs for approximately 9 days. The distillation loss weight λ is tuned based on validation performance to balance action prediction accuracy and geometric alignment.

2) *Stage 2: Posttraining on Downstream Tasks*: After pretraining, we adapt GLaD to specific downstream tasks (e.g., LIBERO) through supervised fine-tuning (Fig. 2(b)). During this stage, the VLA backbone is adapted via LoRA [86] for parameter-efficient fine-tuning, while the action decoder and feature alignment module are fully trained. The VGGT teacher network remains frozen to preserve the learned geometric priors. Task-specific posttraining is conducted for 60k steps with learning rate 3.5e-5 on 8×A100 GPUs.

IV. EXPERIMENTS

We train and evaluate GLaD across three stages: large-scale pretraining on the Bridge dataset, post-training on LIBERO dataset, and comprehensive evaluation on the standard LIBERO benchmark and enhanced LIBERO-PRO benchmark. In Section IV-A, we introduce the datasets and benchmarks used in our experiments. In Section IV-B, we detail our training protocol including pretraining and post-training phases. In Section IV-C, we present evaluation results on the standard LIBERO benchmark, where GLaD achieves state-of-the-art 94.1% average success rate. In Section IV-D, we

discuss the robustness evaluation framework LIBERO-PRO. Finally, in Section IV-E, we conduct ablation studies to analyze the impact of pretraining checkpoint selection and post-training duration.

A. Datasets and Benchmarks

1) *Bridge Dataset*: We use the Bridge dataset [87] for large-scale pretraining. The Bridge dataset provides diverse manipulation demonstrations that help the model acquire foundational visuomotor skills. We choose Bridge over larger datasets like OXE because Bridge alone provides sufficient diversity and scale for our pretraining objectives, while being more computationally efficient.

2) *LIBERO Benchmark and LIBERO Dataset*: LIBERO [9] is a benchmark for lifelong learning in robot manipulation, featuring procedurally generated tasks based on everyday human activities. The benchmark includes 130 language-conditioned manipulation tasks organized into four suites, each designed to evaluate different aspects of knowledge transfer:

LIBERO-Spatial (10 tasks) tests the transfer of spatial knowledge. All tasks require placing a bowl on a plate among the same set of objects, but the bowl’s location varies across tasks. Success requires continually learning and memorizing new spatial relationships.

LIBERO-Object (10 tasks) evaluates object-level knowledge transfer. Each task involves pick-and-place of a unique object, requiring the agent to recognize and manipulate different object types.

LIBERO-Goal (10 tasks) assesses procedural knowledge transfer. All tasks share the same objects and spatial layout but differ in task goals, requiring the agent to learn diverse manipulation behaviors.

LIBERO-Long (10 tasks) contains long-horizon manipulation tasks that combine multiple subtasks, testing the model’s ability to handle complex, multi-step procedures.

Each task is accompanied by 50 high-quality human tele-operation demonstrations. Following LIBERO protocol, we evaluate on 50 episodes per task.

3) *LIBERO-PRO Benchmark*: While LIBERO provides a standardized evaluation framework, recent work [10] has revealed critical limitations: models achieving over 90% success on standard LIBERO often fail completely under minor perturbations, suggesting reliance on memorization rather than genuine task understanding.

To test this, we evaluate on LIBERO-PRO [10], which extends LIBERO with controlled perturbations across four dimensions:

Object Perturbations modify non-essential object attributes (color, texture, size) while preserving semantic equivalence, testing robustness to superficial visual changes.

Position Perturbations alter initial object placements, both absolute positions and relative spatial arrangements, probing spatial reasoning under varied layouts.

Semantic Perturbations rephrase task instructions while preserving the original task intent (e.g., “pick up” → “grab”, “place on” → “put on top of”), evaluating whether the model genuinely understands language semantics or merely pattern-matches specific phrasings.

TABLE I: Results on LIBERO benchmark across four evaluation suites. We compare GLaD against state-of-the-art VLA baselines across spatial reasoning (LIBERO-SPATIAL), object manipulation (LIBERO-OBJECT), goal-oriented tasks (LIBERO-GOAL), and long-horizon procedures (LIBERO-LONG). Success rates (%) are averaged over 50 episodes per task. GLaD achieves competitive performance with 94.1% average success rate, ranking among top-tier methods and demonstrating particularly strong object manipulation capabilities (97.4%, highest among all methods). **Bold** indicates highest performance and underline indicates second-highest performance.

	spatial	object	goal	long	average
lapa	73.8	74.6	58.8	55.4	65.7
diffusion policy	78.3	92.5	68.3	50.5	72.4
octo	78.9	85.7	84.6	51.1	75.1
mdt	78.5	87.5	73.5	64.8	76.1
openvla	84.7	88.4	79.2	53.7	76.5
mail	74.3	90.1	81.8	78.6	81.2
univla	95.2	<u>95.4</u>	<u>91.9</u>	<u>87.5</u>	<u>92.5</u>
GLaD	<u>95</u>	97.4	94.4	89.4	94.1

Task Perturbations modify the task itself by changing target objects or required actions, while ensuring all components (objects and actions) appear in the training set. This tests compositional generalization—the ability to recombine known elements in novel ways.

Unlike the standard LIBERO evaluation where test tasks closely mirror training tasks, LIBERO-PRO introduces sufficient variation to distinguish between memorization and genuine generalization. And for LIBERO-PRO, we focus our evaluation on the perturbation types that are consistently available across all task suites in the current benchmark release.

B. Training Protocol

1) *Pretraining Phase:* We pretrain the model on the Bridge dataset using $8 \times$ A100 GPUs for 9 days, spanning 45 epochs with learning rate $5e-7$. Pretraining enables the model to acquire general visuomotor skills from large-scale diverse data before specializing on LIBERO tasks. We use checkpoints at epochs 27 and 45, both of which serve as initialization for subsequent post-training experiments.

2) *Post-training Phase:* We perform task-specific post-training on the LIBERO dataset to adapt the model to LIBERO’s task distribution. We train for 48k steps with learning rate $3.5e-5$, saving checkpoints every 16k steps. All post-training experiments use $8 \times$ A100 GPUs. We select the best-performing checkpoints based on validation performance for final evaluation.

C. Evaluation on LIBERO Benchmark

Evaluation Setup: We evaluate on all four LIBERO task suites using the standard simulator environment. Following the LIBERO protocol, we report success rates averaged over 50 evaluation episodes per task. We use the data processing pipeline from OpenVLA to exclude failure demonstrations during training.

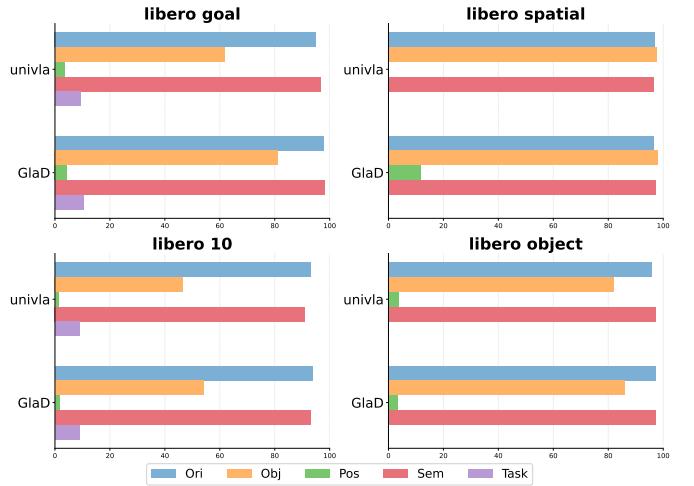


Fig. 3: Robustness comparison across LIBERO suites under five perturbation types. We compare GLaD against UniVLA on four LIBERO suites: GOAL, SPATIAL, Long-horizon (10), and OBJECT. **Ori**: Original tasks; **Obj**: Object perturbations (color, texture, size); **Pos**: Position perturbations; **Sem**: Semantic perturbations (language); **Task**: Task perturbations. Success rates (%) averaged over 50 episodes per task. GLaD demonstrates significant improvements in object perturbation robustness, particularly on GOAL (81% vs 62%) and Long (54% vs 47%).

Results: Table I summarizes the performance across all task suites. Our GLaD, pretrained only on Bridge dataset, achieves 94.1% average success rate, outperforming UniVLA (92.5%) which uses the same pretraining data. This demonstrates that geometry-aware pretraining provides efficiency gains comparable to data scaling. GLaD also substantially outperforms other baselines including MAIL, OpenVLA, MDT, Octo , and Diffusion Policy.

GLaD demonstrates strong performance across three suites: LIBERO-OBJECT (97.4%), LIBERO-GOAL (94.4%), and LIBERO-LONG (89.4%). The particularly strong performance on LIBERO-OBJECT validates that geometry-aware pretraining effectively captures object-level visual features and manipulation affordances. The consistent improvements over UniVLA across all suites demonstrate the effectiveness of incorporating geometric structure into VLA pretraining.

D. Robustness Analysis on LIBERO-PRO

Evaluation Setup: We evaluate GLaD and UniVLA on LIBERO-PRO to assess robustness under controlled perturbations. As introduced in Section IV-A, LIBERO-PRO distinguishes between genuine task understanding and memorization through systematic variations in objects, positions, semantics, and task compositions. Figure 3 visualizes the overall robustness comparison across all perturbation types, while Table II summarizes the averaged results across all four suites. Detailed per-task results are provided in Appendix A.

Results: As shown in Figure 3 and Table II, compared with UniVLA, GLaD demonstrates significant advantages under object perturbations, which modify non-essential visual attributes

TABLE II: Average success rates across four LIBERO suites under different perturbation types. The columns represent: **Ori** (original task without perturbations), **Obj** (object appearance perturbations including color, texture, and size changes), **Pos** (position perturbations with spatial layout variations), **Sem** (semantic perturbations with language rephrasing), and **Task** (task perturbations with compositional changes using known elements). Results are averaged over all tasks in each suite (50 episodes per task). GLaD demonstrates significant advantages under object perturbations: 81% on LIBERO-Goal compared to UniVLA’s 62%, and 54% on LIBERO-10 compared to 47%, showing improved robustness to visual appearance variations. Complete per-task results are provided in Appendix A.

Benchmark	univla					GLaD				
	Ori	Obj	Pos	Sem	Task	Ori	Obj	Pos	Sem	Task
LIBERO-Goal	95	62	4	97	9	98	81	4	98	10
LIBERO-Spatial	97	98	0	97	–	97	98	12	97	–
LIBERO-10	93	47	1	91	9	94	54	2	93	9
LIBERO-Object	96	82	4	97	0	97	86	3	97	0

(color, texture, size) while preserving semantic equivalence. On LIBERO-GOAL, GLaD achieves 81% average success rate compared to UniVLA’s 62%, a substantial +19 percentage point improvement. This gap is even more pronounced in specific tasks: for “Put(bowl, plate)”, GLaD reaches 84% while UniVLA achieves only 24%—a 60 percentage point difference (detailed results in Table IV). On LIBERO-LONG, GLaD achieves 54% compared to UniVLA’s 47% (Table VI), showing improved robustness on long-horizon tasks with appearance variations. On LIBERO-OBJECT, GLaD maintains 86% success rate versus UniVLA’s 82% (Table VII).

These results validate GLaD’s core design principle: geometry-aware pretraining enables the model to learn intrinsic geometric features and manipulation affordances rather than relying on superficial visual characteristics. This proves critical when object appearances change while geometric structure remains constant. The consistent improvements across all four suites demonstrate that geometric understanding generalizes across different task types and complexities.

Both models exhibit strong semantic robustness, achieving 93-98% success rates across all suites under language rephrasing. This demonstrates that VLA architectures with large language model backbones effectively generalize across language variations, understanding task intent despite different phrasings.

On position perturbations, both models show limited robustness. GLaD achieves 12% on LIBERO-SPATIAL compared to UniVLA’s 0% (Table V), suggesting modest improvements in spatial reasoning. However, performance remains low on other suites (1-4%), indicating that spatial layout variations bring substantial challenges. Task perturbations, which test compositional generalization by recombining known elements in novel ways, remain challenging for both approaches with 9-10% success rates. This reveals shared limitations of current VLA methods in handling compositional reasoning and novel task configurations.

Overall, the LIBERO-PRO evaluation demonstrates that GLaD’s geometry-aware pretraining provides substantial robustness advantages specifically in scenarios involving visual appearance variations—precisely the domain where geometric understanding matters most. These findings align with and reinforce the results from standard LIBERO benchmark (Section IV-C), further validating the effectiveness of incorporating

TABLE III: **Ablation study on key architectural design choices.** We evaluate three critical design dimensions: geometry encoder selection (VGTT vs. PI3), feature alignment layer (layer 32 vs. layer 24), and geometry integration strategy (late fusion vs. early weighted fusion). Success rates (%) are averaged over 50 episodes per task across all four LIBERO suites. Bold numbers indicate the best performance in each column. **GLaD (full):** VGTT encoder + Layer 32/32 alignment + Late fusion in LLM representation space. **PI3:** Replaces VGTT with Pi3 encoder (Permutation-Equivariant Visual Geometry Learning encoder). **Layer 24/32:** Aligns geometric features with layer 24 instead of final layer 32. **Weighted Fusion:** Aligns VGTT features with DinoSigLIP features, then performs weighted combination before LLM input (early fusion).

Configuration	Spatial	Object	Goal	Long	Average
GLaD	95.0	97.4	94.4	89.4	94.1
<i>Geometry Encoder Ablation</i>					
PI3	65.2	98.6	94.2	86.4	86.1
<i>Feature Alignment Layer Ablation</i>					
Layer 24/32	94.4	90.6	94.4	91.0	92.6
<i>Geometry Integration Strategy Ablation</i>					
Weighted Fusion	87.6	80.8	91.4	76.0	84.0

geometric structure into vision-language-action models.

E. Ablation Studies

We conduct comprehensive ablation experiments to validate the key design choices in GLaD’s geometry-aware architecture. Table III presents results across three critical dimensions: geometry encoder selection, feature alignment strategy, and geometry integration method. We also provide attention pattern analysis to offer qualitative insights into how these design choices affect task-relevant object localization and manipulation reasoning.

Geometry Encoder Architecture: We compare two geometry encoders: VGTT [8] and PI3 [34]. Notably, LIBERO-SPATIAL exhibits the highest sensitivity to geometry encoder selection across all ablations, with VGTT achieving 95.0% compared to 65.2% with PI3—a 29.8 percentage point difference that represents the largest performance gap among all task suites. This directly validates that VGTT’s geometry-grounded

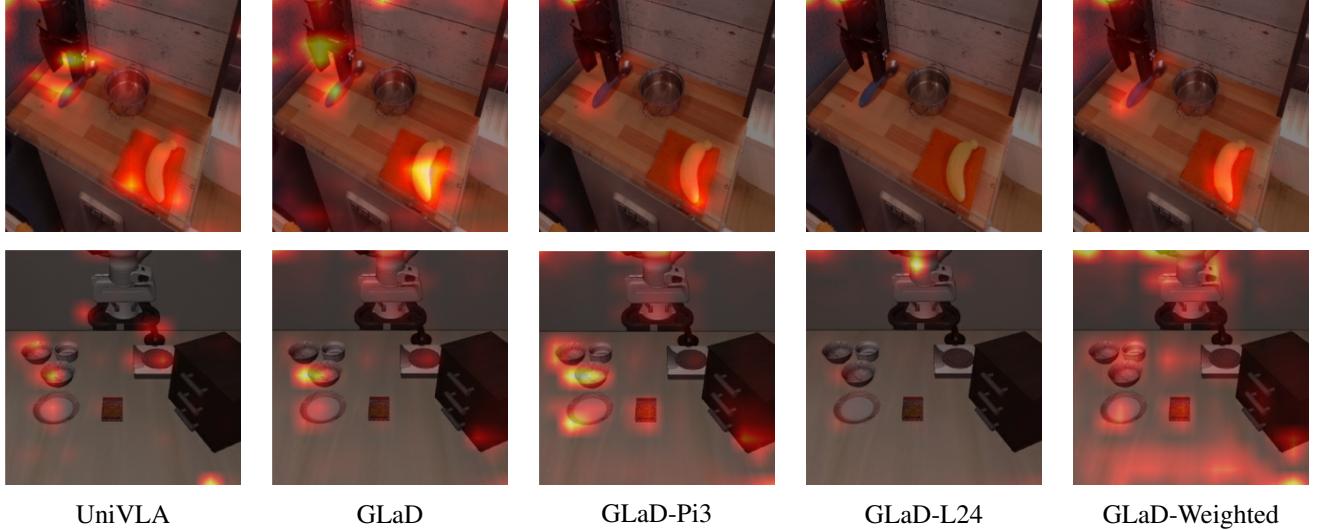


Fig. 4: Attention maps across model variants. Up: Bridge scene (“Put the banana in front of the spoon”). Down: LIBERO scene (“Pick up the black bowl between the plate and the ramekin and place it on the plate”). GLaD-Pi3: w/ Pi3 encoder; GLaD-L24: w/ Layer-24 alignment; GLaD-Weighted: w/ early weighted fusion. See Table III for detailed configurations and quantitative results.

visual representation is particularly effective for spatial reasoning tasks, precisely the capability LIBERO-SPATIAL is designed to test. While both encoders achieve strong performance on object-centric tasks (LIBERO-OBJECT: 97.4% vs. 98.6%), VGGT’s substantial advantage on spatial tasks (94.1% average vs. 86.1%) demonstrates that its geometry-aware features provide robust spatial understanding crucial for manipulation tasks requiring precise spatial relationships, such as “place bowl on plate at specific location”.

Feature Alignment Layer: We investigate the impact of aligning geometric features to different layers of the LLM backbone. Our default configuration aligns to the final layer (32/32), while the ablation variant aligns to an earlier layer (24/32). Early-layer alignment (layer 24) achieves 92.6% average success rate, showing notable performance drop on LIBERO-OBJECT (90.6% vs. 97.4%). This demonstrates that aligning geometric features to the final representation layer is crucial for effective multimodal fusion. Late-layer alignment allows the language model to first process visual-semantic features through most of its depth before integrating geometric information, enabling better preservation of both visual semantics and spatial structure. Early alignment may cause geometric signals to be diluted as they propagate through subsequent transformer layers.

Geometry Integration Strategy: We compare two approaches for incorporating geometric information: (1) our default method that aligns geometry features to the final-layer visual tokens in the LLM’s representation space, and (2) an alternative approach that aligns VGGT features to DinoSigLIP features before LLM input, then performs weighted combination. The weighted-feature fusion approach achieves only 84.0% average success rate with particularly poor performance on LIBERO-OBJECT (80.8%) and LIBERO-LONG (76.0%). This substantial degradation suggests that early fusion in the

visual feature space, before language model processing, fails to leverage the LLM’s capacity for multimodal reasoning. Our late-fusion approach enables the language model to learn task-adaptive integration of geometric and semantic cues, rather than relying on fixed weighted combination.

Attention Pattern Analysis: To provide qualitative insights into the quantitative results above, Figure 4 visualizes attention distributions across model variants, revealing how design choices affect task-relevant object localization. GLaD demonstrates sharp, focused attention on manipulation targets (banana in Bridge scene, target plate in LIBERO scene), correlating with its strong performance (94.1% average). In contrast, GLaD-Pi3 exhibits scattered attention across multiple plates in LIBERO, directly explaining its LIBERO-SPATIAL failure (65.2%); GLaD-L24 shows diffused attention unable to identify task-relevant regions, aligning with its LIBERO-OBJECT degradation (90.6%); UniVLA and GLaD-Weighted attend more to the gripper than target objects, indicating reliance on egocentric visual cues rather than object-centric reasoning, which explains GLaD-Weighted’s poor performance (84.0%). These attention patterns provide qualitative evidence supporting Table III’s quantitative results.

We validate that GLaD’s design choices (VGGT for geometry encoding, final-layer alignment, and late-stage feature integration) work synergistically to achieve strong performance across diverse manipulation scenarios. The 8-10% performance gaps observed in ablations highlight the importance of each component, particularly for spatial reasoning and object manipulation tasks.

V. DISCUSSION

A. Why Geometric Understanding Matters

Analysis of attention maps (Fig. 4) reveals that GLaD develops sharper attention on manipulation-relevant objects

compared to baselines. VLA models trained on 2D vision encoders (CLIP, SigLIP) learn semantic correspondences but struggle to ground semantics in 3D spatial structure. By aligning VGGT geometric features to LLM hidden states, GLaD learns representations capturing both *what* objects are and *what* they look like, proving particularly valuable for object-centric tasks (97.4% on LIBERO-OBJECT).

B. Design Choices and Robustness

Our ablation studies (Table III) show that late-stage alignment to LLM hidden states substantially outperforms early fusion (94.1% vs. 84.0%), enabling task-adaptive integration of geometric and semantic cues. LIBERO-PRO evaluation reveals an asymmetry: GLaD demonstrates strong robustness to object appearance perturbations but limited improvement on position perturbations. This validates our hypothesis—geometric features ground representations in spatial structure rather than superficial appearance, making the model robust when colors or textures change while geometric affordances remain constant.

C. Alternative Approaches and Limitations

We explored explicit geometry supervision (predicting depth maps) and implicit supervision (contrastive learning), but both failed: explicit supervision caused training divergence due to conflicting objectives, while implicit supervision did not outperform baselines. These failures validate our design of aligning pretrained geometry encoder features to LLM hidden states. Limitations remain in position perturbation robustness, though consistent improvements across LIBERO suites suggest geometric understanding is a valuable inductive bias for VLA models.

VI. CONCLUSION

Current vision-language-action models lack geometric understanding due to reliance on 2D vision encoders (CLIP, SigLIP) that do not encode spatial positions and object relations. We proposed GLaD, a geometry-aware VLA framework that incorporates 3D geometric priors during pretraining through knowledge distillation from a frozen Visual Geometry Grounded Transformer (VGGT). Our key contribution is a late-stage feature alignment mechanism that distills geometric features into the LLM’s hidden states corresponding to visual tokens, enabling task-adaptive integration of geometric and semantic cues.

GLaD achieves 94.1% average success rate on LIBERO benchmark, outperforming UniVLA (92.5%) trained on identical data. On LIBERO-PRO robustness benchmark, GLaD demonstrates improved resilience to perturbations, particularly excelling under object appearance variations, validating that geometry-aware pretraining enhances policy generalization beyond superficial pattern matching. Ablation studies confirm that VGGT geometry encoding, final-layer alignment, and late-stage integration each contribute significantly to performance.

While limitations remain in spatial layout generalization, our results establish that incorporating geometric priors during pretraining is a promising direction for building more capable vision-language-action models for robotic manipulation.

REFERENCES

- [1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, “Openvla: An open-source vision-language-action model,” 2024. [Online]. Available: <https://arxiv.org/abs/2406.09246>
- [2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, “ π_0 : A vision-language-action flow model for general robot control,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.24164>
- [3] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky, “ $\pi_{0.5}$: a vision-language-action model with open-world generalization,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.16054>
- [4] P. Intelligence, A. Amin, R. Aniceto, A. Balakrishna, K. Black, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, J. DiCarlo, D. Driess, M. Equi, A. Esmail, Y. Fang, C. Finn, C. Glossop, T. Godden, I. Goryachev, L. Groom, H. Hancock, K. Hausman, G. Hussein, B. Ichter, S. Jakubczak, R. Jen, T. Jones, B. Katz, L. Ke, C. Kuchi, M. Lamb, D. LeBlanc, S. Levine, A. Li-Bell, Y. Lu, V. Mano, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, C. Sharma, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, W. Stoeckle, A. Swerdlow, J. Tanner, M. Torne, Q. Vuong, A. Walling, H. Wang, B. Williams, S. Yoo, L. Yu, U. Zhilinsky, and Z. Zhou, “ $\pi_{0.6}^*$: a vla that learns from experience,” 2025. [Online]. Available: <https://arxiv.org/abs/2511.14759>
- [5] 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. [Online]. Available: <https://arxiv.org/abs/2103.00020>
- [6] M. Oquab, T. Dariseti, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, “Dinov2: Learning robust visual features without supervision,” 2024. [Online]. Available: <https://arxiv.org/abs/2304.07193>
- [7] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, “Sigmoid loss for language image pre-training,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.15343>
- [8] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “Vggt: Visual geometry grounded transformer,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.11651>
- [9] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, “Libero: Benchmarking knowledge transfer for lifelong robot learning,” 2023. [Online]. Available: <https://arxiv.org/abs/2306.03310>
- [10] X. Zhou, Y. Xu, G. Tie, Y. Chen, G. Zhang, D. Chu, P. Zhou, and L. Sun, “Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization,” 2025. [Online]. Available: <https://arxiv.org/abs/2510.03827>
- [11] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li, “Univila: Learning to act anywhere with task-centric latent actions,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.06111>
- [12] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peraita, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitzkovich, “Rt-1: Robotics transformer for real-world control at scale,” 2023. [Online]. Available: <https://arxiv.org/abs/2212.06817>
- [13] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet,

- J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023. [Online]. Available: <https://arxiv.org/abs/2307.15818>
- [14] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," 2024. [Online]. Available: <https://arxiv.org/abs/2405.12213>
- [15] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, H. Zhang, and M. Zhu, "Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation," 2024. [Online]. Available: <https://arxiv.org/abs/2410.06158>
- [16] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang, X. Wang, B. Liu, J. Fu, J. Bao, D. Chen, Y. Shi, J. Yang, and B. Guo, "Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," 2024. [Online]. Available: <https://arxiv.org/abs/2411.19650>
- [17] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," 2025. [Online]. Available: <https://arxiv.org/abs/2410.07864>
- [18] 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," 2024. [Online]. Available: <https://arxiv.org/abs/2303.04137>
- [19] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, and X. Li, "Spatialvla: Exploring spatial representations for visual-language-action model," 2025. [Online]. Available: <https://arxiv.org/abs/2501.15830>
- [20] I. Singh, A. Goyal, S. Birchfield, D. Fox, A. Garg, and V. Blukis, "Og-vla: Orthographic image generation for 3d-aware vision-language action model," 2025. [Online]. Available: <https://arxiv.org/abs/2506.01196>
- [21] C. Li, J. Wen, Y. Peng, Y. Peng, F. Feng, and Y. Zhu, "Pointvla: Injecting the 3d world into vision-language-action models," 2025. [Online]. Available: <https://arxiv.org/abs/2503.07511>
- [22] J. Zhang, Y. Chen, Y. Xu, Z. Huang, Y. Zhou, Y.-J. Yuan, X. Cai, G. Huang, X. Quan, H. Xu, and L. Zhang, "4d-vla: Spatiotemporal vision-language-action pretraining with cross-scene calibration," 2025. [Online]. Available: <https://arxiv.org/abs/2506.22242>
- [23] W. Cai, I. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao, "Spatialbot: Precise spatial understanding with vision language models," 2025. [Online]. Available: <https://arxiv.org/abs/2406.13642>
- [24] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," 2023. [Online]. Available: <https://arxiv.org/abs/2302.12288>
- [25] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan, "3d-vla: A 3d vision-language-action generative world model," 2024. [Online]. Available: <https://arxiv.org/abs/2403.09631>
- [26] L. Sun, B. Xie, Y. Liu, H. Shi, T. Wang, and J. Cao, "Geovla: Empowering 3d representations in vision-language-action models," 2025. [Online]. Available: <https://arxiv.org/abs/2508.09071>
- [27] M. Chen, L. Cui, W. Zhang, H. Zhang, Y. Zhou, X. Li, S. Tang, J. Liu, B. Liao, H. Chen, X. Liu, and P. Wan, "Midas: Multimodal interactive digital-human synthesis via real-time autoregressive video generation," 2025. [Online]. Available: <https://arxiv.org/abs/2508.19320>
- [28] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," 2021. [Online]. Available: <https://arxiv.org/abs/2103.13413>
- [29] G. Bae and A. J. Davison, "Rethinking inductive biases for surface normal estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [30] C. Ye, L. Qiu, X. Gu, Q. Zuo, Y. Wu, Z. Dong, L. Bo, Y. Xiu, and X. Han, "Stablenormal: Reducing diffusion variance for stable and sharp normal," *ACM Transactions on Graphics*, 2024.
- [31] G. Bae, I. Budvytis, and R. Cipolla, "Estimating and exploiting the aleatoric uncertainty in surface normal estimation," in *International Conference on Computer Vision (ICCV)*, 2021.
- [32] R. Fan, H. Wang, B. Xue, H. Huang, Y. Wang, M. Liu, and I. Pitas, "Three-filters-to-normal: An accurate and ultrafast surface normal estimator," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5405–5412, 2021.
- [33] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," 2020. [Online]. Available: <https://arxiv.org/abs/2003.08934>
- [34] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He, " π^3 : Permutation-equivariant visual geometry learning," 2025. [Online]. Available: <https://arxiv.org/abs/2507.13347>
- [35] Z. Xu, Z. Li, Z. Dong, X. Zhou, R. Newcombe, and Z. Lv, "4dgt: Learning a 4d gaussian transformer using real-world monocular videos," 2025.
- [36] Y. Xu, J. Zhang, Z. Huang, Y. Chen, Y. Zhou, C. Zhenyu, Y. Yuan, P. Xia, G. Huang, X. Cai, Z. Qi, X. Quan, J. Hao, H. Xu, and L. Zhang, "Uniggg: Unified 3d understanding and generation via geometric-semantic encoding," *arXiv preprint arXiv:2508.11952*, 2025.
- [37] X. Huang, J. Wu, Q. Xie, and K. Han, "3drs: Mllms need 3d-aware representation supervision for scene understanding," in *NeurIPS*, 2025.
- [38] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *CVPR*, 2024.
- [39] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: <https://arxiv.org/abs/1503.02531>
- [40] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," 2015. [Online]. Available: <https://arxiv.org/abs/1412.6550>
- [41] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," 2017. [Online]. Available: <https://arxiv.org/abs/1612.03928>
- [42] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," 2019. [Online]. Available: <https://arxiv.org/abs/1904.05068>
- [43] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, "A survey on knowledge distillation of large language models," 2024. [Online]. Available: <https://arxiv.org/abs/2402.13116>
- [44] X. He, Z. Lin, Y. Gong, A.-L. Jin, H. Zhang, C. Lin, J. Jiao, S. M. Yiu, N. Duan, and W. Chen, "Annollm: Making large language models to be better crowdsourced annotators," 2024. [Online]. Available: <https://arxiv.org/abs/2303.16854>
- [45] Y. Wang, Z. Yu, Z. Zeng, L. Yang, C. Wang, H. Chen, C. Jiang, R. Xie, J. Wang, X. Xie, W. Ye, S. Zhang, and Y. Zhang, "Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization," 2024. [Online]. Available: <https://arxiv.org/abs/2306.05087>
- [46] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, and A. Awadallah, "Orca: Progressive learning from complex explanation traces of gpt-4," 2023. [Online]. Available: <https://arxiv.org/abs/2306.02707>
- [47] A. Mitra, L. D. Corro, S. Mahajan, A. Codas, C. Simoes, S. Agarwal, X. Chen, A. Razdaibiedina, E. Jones, K. Aggarwal, H. Palangi, G. Zheng, C. Rosset, H. Khanpour, and A. Awadallah, "Orca 2: Teaching small language models how to reason," 2023. [Online]. Available: <https://arxiv.org/abs/2311.11045>
- [48] C. Xu, D. Guo, N. Duan, and J. McAuley, "Baize: An open-source chat model with parameter-efficient tuning on self-chat data," 2023. [Online]. Available: <https://arxiv.org/abs/2304.01196>
- [49] X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen, "Mammoth: Building math generalist models through hybrid instruction tuning," 2023. [Online]. Available: <https://arxiv.org/abs/2309.05653>
- [50] C. Li, Q. Chen, L. Li, C. Wang, Y. Li, Z. Chen, and Y. Zhang, "Mixed distillation helps smaller language model better reasoning," 2024. [Online]. Available: <https://arxiv.org/abs/2312.10730>
- [51] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language models with self-generated instructions," 2023. [Online]. Available: <https://arxiv.org/abs/2212.10560>
- [52] L. Ranaldi, G. Pucci, and A. Freitas, "Empowering cross-lingual abilities of instruction-tuned large language models by translation-following demonstrations," in *Findings of the Association for Computational Linguistics ACL 2024*. Association for Computational Linguistics, 2024, p. 7961–7973. [Online]. Available: <http://dx.doi.org/10.18653/v1/2024.findings-acl.473>
- [53] Z. Sun, Y. Shen, Q. Zhou, H. Zhang, Z. Chen, D. Cox, Y. Yang, and C. Gan, "Principle-driven self-alignment of language models from scratch with minimal human supervision," 2023. [Online]. Available: <https://arxiv.org/abs/2305.03047>
- [54] Z. Luo, C. Xu, P. Zhao, Q. Sun, X. Geng, W. Hu, C. Tao, J. Ma, Q. Lin, and D. Jiang, "Wizardcoder: Empowering code large language models with evol-instruct," 2025. [Online]. Available: <https://arxiv.org/abs/2306.08568>
- [55] H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, Y. Tang, and D. Zhang, "Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct," 2025. [Online]. Available: <https://arxiv.org/abs/2308.09583>

- [56] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, Q. Lin, and D. Jiang, “Wizardlm: Empowering large pre-trained language models to follow complex instructions,” 2025. [Online]. Available: <https://arxiv.org/abs/2304.12244>
- [57] H. Dai, Z. Liu, W. Liao, X. Huang, Y. Cao, Z. Wu, L. Zhao, S. Xu, W. Liu, N. Liu, S. Li, D. Zhu, H. Cai, L. Sun, Q. Li, D. Shen, T. Liu, and X. Li, “Auggpt: Leveraging chatgpt for text data augmentation,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.13007>
- [58] N. Ding, Y. Chen, B. Xu, Y. Qin, Z. Zheng, S. Hu, Z. Liu, M. Sun, and B. Zhou, “Enhancing chat language models by scaling high-quality instructional conversations,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.14233>
- [59] S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. D. Giorno, S. Gopi, M. Javaheripir, P. Kauffmann, G. de Rosa, O. Saarikivi, A. Salim, S. Shah, H. S. Behl, X. Wang, S. Bubeck, R. Eldan, A. T. Kalai, Y. T. Lee, and Y. Li, “Textbooks are all you need,” 2023. [Online]. Available: <https://arxiv.org/abs/2306.11644>
- [60] Y. Li, S. Bubeck, R. Eldan, A. D. Giorno, S. Gunasekar, and Y. T. Lee, “Textbooks are all you need ii: phi-1.5 technical report,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.05463>
- [61] Y. Wei, Z. Wang, J. Liu, Y. Ding, and L. Zhang, “Magicoder: Empowering code generation with oss-instruct,” 2024. [Online]. Available: <https://arxiv.org/abs/2312.02120>
- [62] Z. Yu, X. Zhang, N. Shang, Y. Huang, C. Xu, Y. Zhao, W. Hu, and Q. Yin, “Wavecoder: Widespread and versatile enhancement for code large language models by instruction tuning,” 2024. [Online]. Available: <https://arxiv.org/abs/2312.14187>
- [63] J. Ye, J. Gao, Q. Li, H. Xu, J. Feng, Z. Wu, T. Yu, and L. Kong, “Zerogen: Efficient zero-shot learning via dataset generation,” 2022. [Online]. Available: <https://arxiv.org/abs/2202.07922>
- [64] J. Gao, R. Pi, Y. Lin, H. Xu, J. Ye, Z. Wu, W. Zhang, X. Liang, Z. Li, and L. Kong, “Self-guided noise-free data generation for efficient zero-shot learning,” 2023. [Online]. Available: <https://arxiv.org/abs/2205.12679>
- [65] L. Bonifacio, H. Abonizio, M. Fadaee, and R. Nogueira, “Inpars: Data augmentation for information retrieval using large language models,” 2022. [Online]. Available: <https://arxiv.org/abs/2202.05144>
- [66] I. Timiryasov and J.-L. Tastet, “Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty,” 2023. [Online]. Available: <https://arxiv.org/abs/2308.02019>
- [67] Y. Gu, L. Dong, F. Wei, and M. Huang, “Minillm: Knowledge distillation of large language models,” 2025. [Online]. Available: <https://arxiv.org/abs/2306.08543>
- [68] Z. Liu, B. Oguz, C. Zhao, E. Chang, P. Stock, Y. Mehdad, Y. Shi, R. Krishnamoorthi, and V. Chandra, “Llm-qat: Data-free quantization aware training for large language models,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.17888>
- [69] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S. E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conery, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan, “Constitutional ai: Harmlessness from ai feedback,” 2022. [Online]. Available: <https://arxiv.org/abs/2212.08073>
- [70] G. Cui, L. Yuan, N. Ding, G. Yao, B. He, W. Zhu, Y. Ni, G. Xie, R. Xie, Y. Lin, Z. Liu, and M. Sun, “Ultrafeedback: Boosting language models with scaled ai feedback,” 2024. [Online]. Available: <https://arxiv.org/abs/2310.01377>
- [71] H. Chen, A. Saha, S. Hoi, and S. Joty, “Personalized distillation: Empowering open-sourced LLMs with adaptive learning for code generation,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 6737–6749. [Online]. Available: <https://aclanthology.org/2023.emnlp-main.417/>
- [72] L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, N. Sarrazin, O. Sanseviero, A. M. Rush, and T. Wolf, “Zephyr: Direct distillation of lm alignment,” 2023. [Online]. Available: <https://arxiv.org/abs/2310.16944>
- [73] J. Hong, Q. Tu, C. Chen, X. Gao, J. Zhang, and R. Yan, “Cyclealign: Iterative distillation from black-box lm to white-box models for better human alignment,” 2023. [Online]. Available: <https://arxiv.org/abs/2310.16271>
- [74] H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi, and S. Prakash, “Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback,” 2024. [Online]. Available: <https://arxiv.org/abs/2309.00267>
- [75] Y. Jiang, C. Chan, M. Chen, and W. Wang, “Lion: Adversarial distillation of proprietary large language models,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.12870>
- [76] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, “Self-instruct: Aligning language models with self-generated instructions,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 13 484–13 508. [Online]. Available: <https://aclanthology.org/2023.acl-long.754/>
- [77] K. Yang, D. Klein, A. Celikyilmaz, N. Peng, and Y. Tian, “Rlcld: Reinforcement learning from contrastive distillation for language model alignment,” 2024. [Online]. Available: <https://arxiv.org/abs/2307.12950>
- [78] J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han, “Large language models can self-improve,” 2022. [Online]. Available: <https://arxiv.org/abs/2210.11610>
- [79] C. Gulcehre, T. L. Paine, S. Srinivasan, K. Konyushkova, L. Weerts, A. Sharma, A. Siddhant, A. Ahern, M. Wang, C. Gu, W. Macherey, A. Doucet, O. Firat, and N. de Freitas, “Reinforced self-training (rest) for language modeling,” 2023. [Online]. Available: <https://arxiv.org/abs/2308.08998>
- [80] W. Yuan, R. Y. Pang, K. Cho, X. Li, S. Sukhbaatar, J. Xu, and J. Weston, “Self-rewarding language models,” 2025. [Online]. Available: <https://arxiv.org/abs/2401.10020>
- [81] E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman, “Star: Bootstrapping reasoning with reasoning,” 2022. [Online]. Available: <https://arxiv.org/abs/2203.14465>
- [82] F. Li, W. Song, H. Zhao, J. Wang, P. Ding, D. Wang, L. Zeng, and H. Li, “Spatial forcing: Implicit spatial representation alignment for vision-language-action model,” 2025. [Online]. Available: <https://arxiv.org/abs/2510.12276>
- [83] H. Chen, B. Xu, S. Zhang, H. Liu, J. Lin, and J. Wang, “Vid-lm: A compact video-based 3d multimodal lm with reconstruction-reasoning synergy,” 2025. [Online]. Available: <https://arxiv.org/abs/2509.24385>
- [84] S. Lee, J. Choi, I. Kang, J. Kim, J. Park, and H. Shim, “3d-aware vision-language models fine-tuning with geometric distillation,” 2025. [Online]. Available: <https://arxiv.org/abs/2506.09883>
- [85] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, “Llama 2: Open foundation and fine-tuned chat models,” 2023. [Online]. Available: <https://arxiv.org/abs/2307.09288>
- [86] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” 2021. [Online]. Available: <https://arxiv.org/abs/2106.09685>
- [87] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, “Bridge data: Boosting generalization of robotic skills with cross-domain datasets,” 2021. [Online]. Available: <https://arxiv.org/abs/2109.13396>

TABLE IV: Detailed per-task results on LIBERO-Goal benchmark. Shows success rates (%) for each task under five perturbation types: original (Ori), object appearance (Obj), position (Pos), semantic/language (Sem), and task composition (Task). Results averaged over 50 episodes per task.

Task	univla					GLaD				
	Ori	Obj	Pos	Sem	Task	Ori	Obj	Pos	Sem	Task
Put(bowl, stove)	92	54	0	98	0	100	96	18	100	0
Put(wine_bottle, cabinet_top)	90	80	0	86	0	98	92	0	94	0
Open(cabinet, drawer_mid)	98	98	0	96	0	96	50	0	98	2
TurnOn(stove)	100	100	0	100	92	100	100	0	100	100
Put(wine_bottle, rack)	90	94	0	96	0	92	88	0	94	0
Open(drawer_top) \wedge Put(bowl, drawer_top)	88	20	36	90	0	96	62	26	98	0
Push(plate, stove_front)	98	100	0	100	0	100	100	0	100	2
Put(bowl, plate)	98	24	0	100	0	98	84	0	100	0
Put(bowl, cabinet_top)	96	18	0	100	0	98	76	0	100	0
Put(cream_cheese, bowl)	100	30	0	100	0	100	64	0	98	0
Average	95	62	4	97	9	98	81	4	98	10

TABLE V: Detailed per-task results on LIBERO-Spatial benchmark. Shows success rates (%) for spatial reasoning tasks. Results averaged over 50 episodes per task.

Task	univla					GLaD				
	Ori	Obj	Pos	Sem	Task	Ori	Obj	Pos	Sem	Task
Pick(on(cookie_box), plate)	100	100	0	100	–	100	100	0	100	–
Pick(next_to(ramekin), plate)	98	100	0	100	–	100	100	0	100	–
Pick(table_center, plate)	98	100	0	100	–	98	100	90	96	–
Pick(between(plate, ramekin), plate)	98	92	0	90	–	100	94	0	90	–
Pick(drawer_top, plate)	92	96	0	94	–	92	100	30	100	–
Pick(next_to(cookie_box), plate)	100	100	0	100	–	100	100	0	100	–
Pick(next_to(plate), plate)	100	92	0	88	–	100	92	0	90	–
Pick(on(ramekin), plate)	94	100	0	98	–	98	100	0	100	–
Pick(on(stove), plate)	94	100	0	100	–	92	100	0	100	–
Pick(on(cabinet), plate)	96	98	0	96	–	88	96	0	98	–
Average	97	98	0	97	–	97	98	12	97	–

TABLE VI: Detailed per-task results on LIBERO-10 benchmark. Shows success rates (%) for long-horizon multi-step manipulation tasks requiring complex sequences with multiple sub-goals. Results averaged over 50 episodes per task.

Task	univla					GLaD				
	Ori	Obj	Pos	Sem	Task	Ori	Obj	Pos	Sem	Task
Put(alphabet_soup, tomato_sauce, basket)	98	0	0	94	0	96	2	0	98	0
TurnOn(stove) \wedge Put(moka_pot, stove)	100	92	0	98	0	96	100	0	96	0
Put(white_mug, plate) \wedge Put(chocolate_pudding, right_of(plate))	88	42	0	88	0	96	70	0	94	0
Put(white_mug, left_plate) \wedge Put(yellow_white_mug, right_plate)	82	74	0	74	0	86	92	0	88	0
Put(black_bowl, drawer_bottom) \wedge Close(drawer_bottom)	96	88	0	92	0	96	86	0	96	0
Put(cream_cheese, butter, basket)	96	96	0	98	0	96	100	0	100	0
Put(alphabet_soup, cream_cheese, basket)	100	0	0	98	0	98	0	0	100	0
Put(moka_pot_1, moka_pot_2, stove)	78	58	0	72	88	78	76	0	70	88
Place(book, caddy_back)	96	0	12	98	0	100	0	18	100	0
Put(yellow_white_mug, microwave) \wedge Close(microwave)	96	16	0	98	0	94	14	0	88	0
Average	93	47	1	91	9	94	54	2	93	9

APPENDIX

A. Detailed LIBERO-PRO Results

This appendix provides complete per-task results for the LIBERO-PRO benchmark evaluation discussed in Section IV-D. While the main text presents averaged success rates across all tasks in each suite (Table II), the detailed tables below show individual task performance under each perturbation type. These results demonstrate the robustness characteristics of GLaD and UniVLA across specific manipulation scenarios, with particular emphasis on object appearance perturbations where GLaD shows significant advantages.

TABLE VII: Detailed per-task results on LIBERO-Object benchmark. Shows success rates (%) for object generalization tasks, testing placement of 10 different objects with varying visual properties. Results averaged over 50 episodes per task.

Task	univla					GlaD				
	Ori	Obj	Pos	Sem	Task	Ori	Obj	Pos	Sem	Task
Place(alphabet_soup, basket)	92	8	0	98	0	100	26	0	98	0
Place(bbq_sauce, basket)	94	70	0	98	0	92	98	12	84	0
Place(butter, basket)	98	96	4	94	0	96	100	0	100	0
Place(chocolate_pudding, basket)	100	98	0	100	0	100	100	0	100	0
Place(cream_cheese, basket)	94	100	36	98	0	98	100	22	100	0
Place(ketchup, basket)	98	74	0	98	0	98	38	0	98	0
Place(milk, basket)	98	78	0	100	0	96	98	0	94	0
Place(orange_juice, basket)	88	98	0	96	0	100	100	0	100	0
Place(salad_dressing, basket)	96	98	0	96	0	100	100	0	98	0
Place(tomato_sauce, basket)	100	100	0	96	0	92	100	0	100	0
Average	96	82	4	97	0	97	86	3	97	0