BridgeVLA: Input-Output Alignment for Efficient 3D Manipulation Learning with Vision-Language Models

Peiyan Li[1,2,3,†], **Yixiang Chen**[1,3], **Hongtao Wu**[2,†,*], **Xiao Ma**[2,†], **Xiangnan Wu**[1]
Yan Huang[1,3,4], **Liang Wang**[1,3], **Tao Kong**[2], **Tieniu Tan**[1,3,5,*]
[1]CASIA [2]ByteDance Seed [3]UCAS [4]FiveAges [5]NJU

[†]Project Lead, [*]Corresponding Author

Abstract

Recently, leveraging pre-trained vision-language models (VLMs) for building vision-language-action (VLA) models has emerged as a promising approach to effective robot manipulation learning. However, only few methods incorporate 3D signals into VLMs for action prediction, and they do not fully leverage the spatial structure inherent in 3D data, leading to low sample efficiency. In this paper, we introduce BridgeVLA, a novel 3D VLA model that (1) projects 3D inputs to multiple 2D images, ensuring input alignment with the VLM backbone, and (2) utilizes 2D heatmaps for action prediction, unifying the input and output spaces within a consistent 2D image space. In addition, we propose a scalable pre-training method that equips the VLM backbone with the capability to predict 2D heatmaps before downstream policy learning. Extensive experiments show the proposed method is able to learn 3D manipulation efficiently and effectively. BridgeVLA outperforms state-of-the-art baseline methods across three simulation benchmarks. In RLBench, it improves the average success rate from 81.4% to 88.2%. In COLOSSEUM, it demonstrates significantly better performance in challenging generalization settings, boosting the average success rate from 56.7% to 64.0%. In GemBench, it surpasses all the comparing baseline methods in terms of average success rate. In real-robot experiments, BridgeVLA outperforms a state-of-the-art baseline method by 32% on average. It generalizes robustly in multiple out-of-distribution settings, including visual disturbances and unseen instructions. Remarkably, it is able to achieve a success rate of 96.8% on 10+ tasks with only 3 trajectories per task, highlighting its extraordinary sample efficiency.

Project Page: https://bridgevla.github.io/
Corresponding Email: wuhongtao.123@bytedance.com; tnt@nlpr.ia.ac.cn

1 Introduction

Leveraging pre-trained vision-language models (VLMs) [1–4] for developing large vision-language-action (VLA) models has become a promising method for learning generalizable and robust manipulation policies [5–9]. However, most VLA models only incorporate 2D image inputs and require extensive efforts on data collection. On the other hand, 3D robot policies leverage 3D structural priors in model design and demonstrate exceptional sample efficiency in learning complex 3D robot manipulation tasks [10–14]. Can we develop a unified 3D VLA model which combines the effectiveness of VLA models with the efficiency from 3D policies?

Although there have been some works exploring integrating 3D information into VLMs for developing 3D VLA

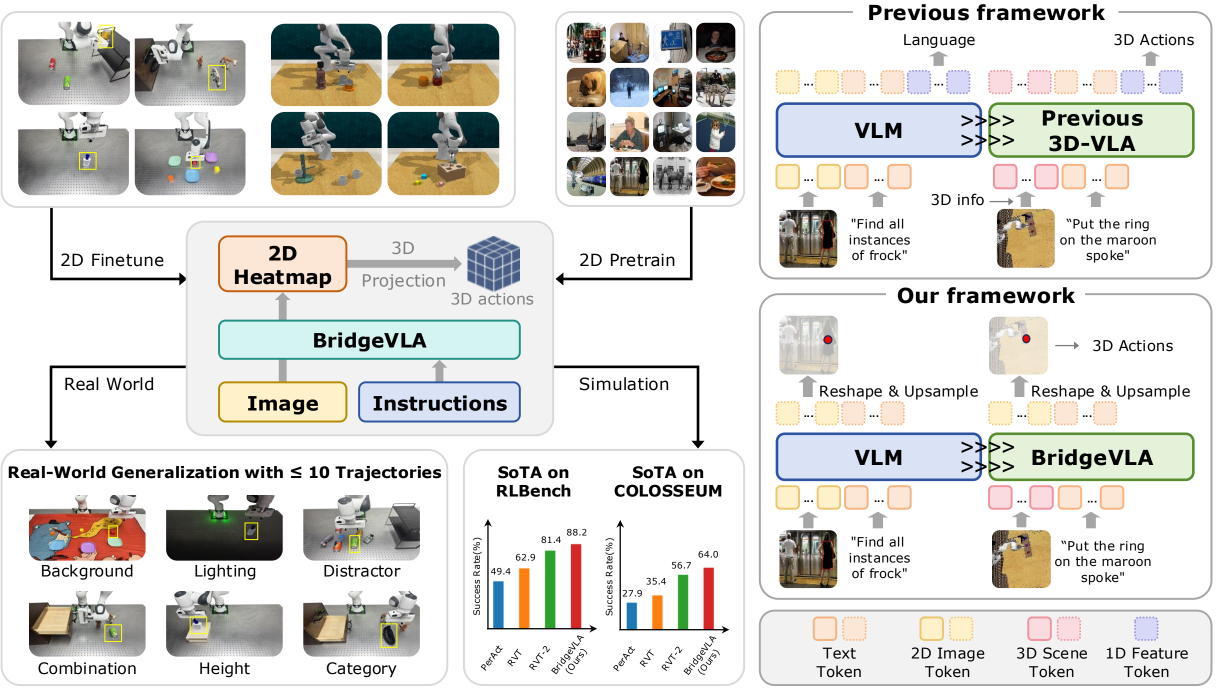


Figure 1 Overview. BridgeVLA is a novel 3D VLA model that aligns the input and output within a unified 2D image space. It is pre-trained on object grounding using 2D heatmaps and fine-tuned on action prediction for 3D manipulation. Experiment results in both simulation and the real world show that it is able to learn 3D manipulation both efficiently and effectively.

models [15, 16], these works typically convert actions into token sequences that do not have spatial structure and use next-token prediction to predict actions. This strategy fails to take advantage of the 3D structural priors as previous efficient 3D policies [10–14] that align the observation input and action output into a unified space, therefore leading to poor sample efficiency. Another significant challenge in developing 3D VLA models lies in the misalignment between the 3D inputs used in action fine-tuning and the 2D image inputs used in original VLM pre-training, causing a large distributional shift from the original VLM pre-training.

To tackle the challenges mentioned above, as illustrated in Fig. 1, we present BridgeVLA, a novel 3D VLA model that achieves remarkable sample efficiency and strong generalization capabilities. To ensure input alignment with the pre-trained VLM backbone, BridgeVLA transforms a 3D point cloud observation into multiple 2D images captured from different orthographic projection views [13, 14]. To leverage the structural priors of the 3D input, BridgeVLA is trained to predict 2D heatmaps for translational action prediction. The 2D heatmaps, generated from the tokens corresponding to the projection images, share the same resolution as these images, aligning the input observations and output actions within a unified spatial structure. Given that the original VLM is pre-trained to predict token sequences, which is incompatible with our VLA's 2D heatmap output, we also introduce a scalable pre-training method, which trains the model to ground objects with heatmaps conditioned on text inputs. This pre-training method equips the VLM with the capabilities to predict heatmaps before downstream fine-tuning for policy learning. **Overall, our design aligns the input and output within a shared 2D space in both pre-training and fine-tuning.**

We perform extensive experiments in both simulation and the real world to evaluate the proposed method. Results show that BridgeVLA is able to learn 3D manipulation both efficiently and effectively. It outperforms state-of-the-art baseline methods in RLBench [17], improving the average success rate from 81.4% to 88.2%. In COLOSSEUM [18], it showcases strong performance in challenging generalization settings, boosting the success rate from 56.7% to 64.0%. In GemBench [19], it surpasses all the comparing baseline methods in terms of average success rate. In real-robot experiments, we evaluate on seven different settings, spanning from visual perturbations to manipulating objects from unseen categories. BridgeVLA surpasses a state-of-the-art method by 32% on average and demonstrates strong performance in generalizing to multiple out-of-distribution settings. Notably, BridgeVLA is able to achieve a success rate of 96.8% on 10+ tasks using only 3 trajectories per task for training, highlighting its superb sample efficiency. In summary, the contributions of this paper are threefold:

- We introduce BridgeVLA, a novel 3D VLA model that efficiently and effectively learns 3D robot manipulation with a vision-language model via input-output alignment with 2D heatmaps.
- We propose a scalable pre-training method to equip the model with the capability to predict heatmaps conditioned on text inputs via object grounding.
- We conduct extensive experiments in both simulation and real-world environments to thoroughly evaluate the proposed method. Results show that BridgeVLA outperforms state-of-the-art methods in both settings and achieves exceptional sample efficiency in real-robot experiments.

2 Related Work

Language-Conditioned Visuomotor Policies. Most language-conditioned visuomotor policies employ transformers to process 2D visual inputs and directly generate 3D actions for manipulation [5–9, 20–25]. In these works, leveraging pre-trained vision-language models (VLMs) for developing large vision-language-action (VLA) models has become popular for its effectiveness on learning complex manipulation [5–9]. However, such 2D image-based policies typically require significant efforts on data collection, often needing hundreds of trajectories per task to learn effectively. On the other hand, 3D manipulation policies hold great potential for efficient learning by taking advantage of the spatial structure inherent in the 3D inputs. A popular line of works take as inputs point cloud data [11, 12, 26–28]. For example, Act3D [12] proposes to create a 3D feature cloud by lifting image features to the observation point cloud and predicts translational actions via classification for 3D points in the observation space. Another line of works utilize voxels to represent the observation space and predict translational actions within the voxel space, unifying the input observation and output actions within the same space [10, 29]. Recently, RVT [13] and RVT-2 [14] propose to leverage orthographic projection of 3D point clouds to convert 3D signals to 2D images to avoid high computational cost on processing 3D inputs. Different from the above methods, our method aims to unify the effectiveness of VLA models and the efficiency of 3D policies within a single cohesive framework, combining the best of both worlds.

3D Vision-Language-Action (VLA) Models. While 2D VLA models have been extensively studied, 3D VLA models [15, 28, 30, 31] remain relatively under-explored. Zhen *et al.* [15] build 3D-VLA on top of a large language model (LLM) and train the model to perform 3D reasoning, multi-modal goal generation, and robot planning. Lift3D [30] proposes to enhance 2D foundation models (*e.g.*, DINOv2 [32]) with implicit and explicit 3D robotic representation for learning 3D manipulation policies. FP3 [28] leverages a transformer to fuse the information from point clouds, proprioceptive states, and language instructions. PointVLA [31] utilizes a VLM and a point cloud encoder to process 2D images and 3D point clouds, respectively. The embeddings from both encoders are injected into an action expert for action prediction. SpatialVLA [16] introduces Ego3D position encoding to inject 3D information into 2D image observation and adaptive action grids to represent robot movement in a more transferable way. Our method is different from the above methods in that it is designed in a way to take advantage of the spatial structure of 3D inputs in action prediction. In addition, it bridges the gap between the 2D image inputs of pre-trained VLMs and the 3D inputs by projecting the 3D inputs into multiple 2D images instead of injecting 3D information into the VLMs. Such design enables it to simultaneously leverages the broad knowledge in the VLM backbone and the spatial structure priors embedded in 3D inputs.

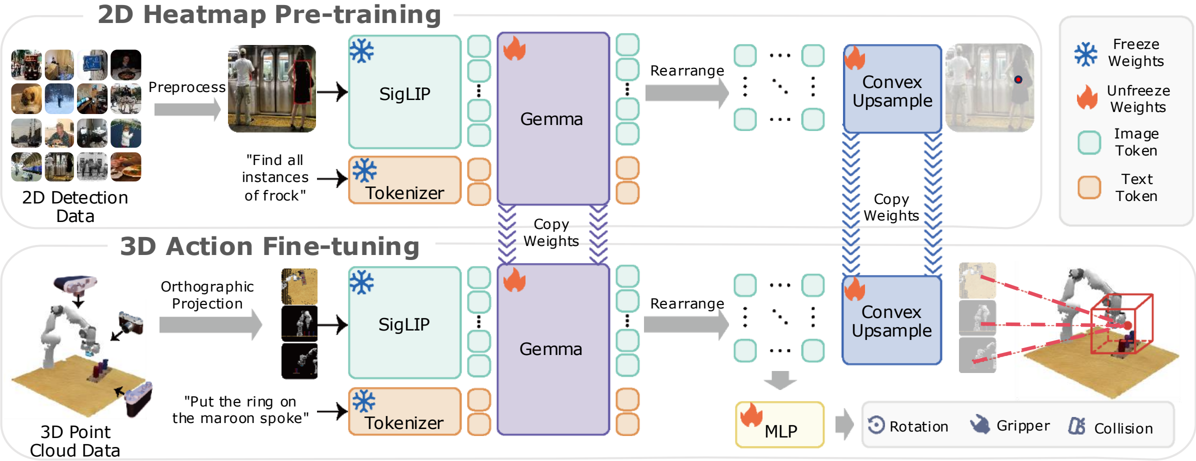


Figure 2 Model Architecture. (a) **2D Heatmap Pre-training:** we train BridgeVLA on 2D object detection datasets. The model takes as inputs an image and a language describing the target object and outputs a 2D heatmap which highlights regions of interest that correspond to the target object. Note that the bounding box shown here is for illustrative purposes only; it is not present in the image when input to the model. (b) **3D Action Fine-tuning:** the model takes as inputs three orthographic projection images of a 3D point cloud and a language instruction. It outputs three 2D heatmaps, which highlight the position of the end-effector in the next keyframe across all three views. For the remaining action components, it uses an MLP to process the image feature tokens to predict the rotation action, gripper action, and collision flag of the next keyframe.

3 BridgeVLA

3.1 Preliminaries

BridgeVLA aims to learn a multi-task 3D robot manipulation policy π, which maps the observation $\mathbf{o}$ and a language instruction l to an action $\mathbf{a}$:

$$\pi : (\mathbf{o}, l) \mapsto \mathbf{a} \tag{1}$$

We assume access to a set of expert demonstrations $\mathcal{D} = \{\tau^i\}_{i=1}^N$ containing N trajectories. And each trajectory contains a language instruction and a sequence of observation-action pairs, *i.e.*, $\tau^i = \{l^i, (\mathbf{o}_1^i, \mathbf{a}_1^i), ..., (\mathbf{o}_H^i, \mathbf{a}_H^i)\}$. The observation $\mathbf{o}$ is one or multiple RGB-D images captured from one or multiple viewpoints. Following prior works [10, 12, 13], the action $\mathbf{a}$ consists of a 6-DoF end-effector pose $T \in SE(3)$, a target gripper state $g \in \{0, 1\}$, and a collision flag $c \in \{0, 1\}$ of the next key frame. The collision flag c indicates whether the motion planner should avoid collisions while moving towards the target pose. A key frame typically captures important or bottleneck steps in a trajectory (detailed in appendix B.1) [33]. BridgeVLA operates through an iterative process: 1) predicting the action $\mathbf{a}_t$ conditioned on the current observation $\mathbf{o}_t$ and instruction l, 2) moving to the predicted next keyframe pose T_t using a sampling-based motion planner [34–36], 3) updating observation and repeating until task completion or reaching a maximum step $H_{\max}$.

As illustrated in Fig. 2, BridgeVLA employs a dual-phase training recipe. During pre-training, it is trained to predict 2D heatmaps on object detection datasets. During fine-tuning, point clouds are projected into multiple 2D images as inputs to the VLM backbone. The model is trained to predict 2D heatmaps for estimating the translational action and other action components. **This design aligns the input and output within a shared 2D space in both pre-training and fine-tuning.**

3.2 2D-Heatmap Pre-training

The VLM backbone was originally pre-trained to predict token sequences without spatial structure. To equip it with the same ability to predict heatmaps as downstream policy learning, we introduce a pre-training stage which trains the model to ground target objects via heatmaps. Concretely, we leverage the 120K object detection split of RoboPoint [37] as our pre-training dataset. For each image, we construct the ground-truth heatmap H^{gt} from the bounding boxes of all objects of interest. Specifically, for each object, we construct a probability map with spatial truncation:

$$H_i^{\text{gt}}(\mathbf{x}) = \begin{cases} p_i(\mathbf{x}) & \text{if } p_i(\mathbf{x}) \geq p_{\min} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\mathbf{x} = (u, v)$ denotes the pixel position, $p_i(\mathbf{x}) = \exp\left(-\|\mathbf{x} - \widehat{\mathbf{x}}_i\|^2 / 2\sigma^2\right)$, $\widehat{\mathbf{x}}_i$ is the center of the object bounding box, and $p_{\min}$ is a probability threshold. For all the objects of interest, we fuse the probability map of all objects via averaging and normalization to obtain H^{gt}:

$$H^{\text{gt}}(\mathbf{x}) = \frac{H_{\text{avg}}(\mathbf{x})}{\sum_{\mathbf{x} \in \Omega} H_{\text{avg}}(\mathbf{x})}, \text{ where } H_{\text{avg}}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} H_i^{\text{gt}}(\mathbf{x}) \tag{3}$$

where Ω denotes the pixel space. Please refer to Fig. 9 for samples of the ground-truth heatmaps.

As illustrated in Fig. 2, we input an image along with the text prompt describing the objects of interest into the VLM backbone of BridgeVLA. In this paper, we employ PaliGemma [1] as the VLM backbone, which consists of a SigLIP vision encoder [38] and a Gemma transformer backbone [39]. During its pre-training, PaliGemma takes as input one or multiple 2D images together with a prefix text (*e.g.*, a question about the image) and outputs a suffix text (*e.g.*, an answer to the question). While the model uses causal attention for predicting suffix text tokens, it adopts bidirectional attention for the image tokens and the prefix text tokens. This allows the image tokens to fuse information from the prefix text.

To predict the heatmap, we first rearrange the output image tokens according to their patch positions to reconstruct the spatial feature grid. A convex upsampling block [40] then converts the grid into a heatmap with the same resolution as the input image. The model is trained with a cross-entropy loss to predict heatmaps that localize the position of all objects of interest in the image. We emphasize that the proposed pre-training strategy outputs a spatially aware 2D heatmap, in contrast to the conventional next-token-prediction used in prior works [15, 16]. Moreover, this approach is highly scalable, as it can, in principle, leverage any vision-language datasets that can be formulated as a heatmap prediction tasks, such as keypoint detection and semantic segmentation.

3.3 3D Action Fine-tuning

During fine-tuning, we first reconstruct a point cloud of the scene from the RGB-D images captured from calibrated cameras. To align with the 2D image input of the VLM backbone, we render three orthographic projection images of the point cloud from three viewpoints (top, front, and right) and use these images as the input images for the VLM backbone as in RVT [13] and RVT-2 [14]. These images, along with the task instruction, are then fed into the pre-trained VLM backbone to generate a heatmap for each of the three views. Importantly, we do not incorporate any additional information (*e.g.*, robot states) during the VLM forward pass to minimize the distribution shift between pre-training and fine-tuning.

For translational actions, we back-project the heatmaps of all three views to estimate the scores of all 3D point grids distributed uniformly across the robot workspace. The position of the 3D point with the highest score determines the end-effector's translation in the next keyframe. Similar to previous works [13, 14], we use Euler angles to represent rotational actions where each axis is discretized into 72 bins. To predict the rotation, binary gripper action, and collision avoidance flag, we integrate features from global and local contexts. For the global feature, max-pooling is applied to the output tokens of each inputted orthographic projection image, resulting in three tokens in total – one for each view. For the local feature, we extract a token from the heatmap peak of each view, also resulting in three tokens in total. All these tokens are concatenated and passed through MLP to predict the rotation action, gripper action, and collision avoidance flag.

Following the approach in prior works [14, 29], BridgeVLA adopts a coarse-to-fine refinement strategy for accurate action prediction. After the initial prediction on the original point cloud, we zoom in and crop the point cloud with a cuboid centered at the predicted translation. A second forward pass is performed on the cropped, zoomed-in point cloud. The predicted action from the second pass is used for execution.

	Overall		Task Success Rate (%)							
Models	**Avg. SR (%) ↑**	**Avg. Rank ↓**	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Open Drawer	Place Cups	Place Wine	Push Buttons
Image-BC (CNN) [10, 41]	1.3	9.3	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0
Image-BC (ViT) [10, 41]	1.3	9.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2F-ARM-BC [10, 29]	20.1	8.7	24.0	24.0	4.0	20.0	20.0	0.0	8.0	72.0
HiveFormer [42]	45.3	6.9	52.0	76.0	0.0	**100.0**	52.0	0.0	80.0	84.0
PolarNet [26]	46.4	6.5	36.0	92.0	4.0	**100.0**	84.0	0.0	40.0	96.0
PerAct [43]	49.4	6.3	55.2±4.7	89.6±4.1	5.6±4.1	70.4±2.0	88.0±5.7	2.4±3.2	44.8±7.8	92.8±3.0
Act3D [12]	65.0	4.3	92.0	92.0	27.0	94.0	93.0	3.0	80.0	99.0
RVT [13]	62.9	4.4	52.0±2.5	99.2±1.6	11.2±3.0	88.0±2.5	71.2±6.9	4.0±2.5	91.0±5.2	**100.0±0.0**
3D Diffuser Actor [11]	81.3	2.5	96.0±2.5	**100.0±0.0**	65.6±4.1	96.8±1.6	89.6±4.1	24.0±7.6	93.6±4.8	98.4±2.0
RVT-2 [14]	81.4	2.5	**100.0±0.0**	99.0±1.7	40.0±0.0	99.0±1.7	74.0±11.8	38.0±4.5	**95.0±3.3**	**100.0±0.0**
BridgeVLA (Ours)	**88.2**	**1.9**	**100.0±0.0**	**100.0±0.0**	**88.0±2.8**	**100.0±0.0**	**100.0±0.0**	**58.4±10.0**	88.0±2.8	98.4±2.2
Models	Put in Cupboard	Put in Drawer	Put in Safe	Screw Bulb	Slide Block	Sort Shape	Stack Blocks	Stack Cups	Sweep to Dustpan	Turn Tap
Image-BC (CNN) [10, 41]	0.0	8.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0
Image-BC (ViT) [10, 41]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.0
C2F-ARM-BC [10, 29]	0.0	4.0	12.0	8.0	16.0	8.0	0.0	0.0	0.0	68.0
HiveFormer [42]	32.0	68.0	76.0	8.0	64.0	8.0	8.0	0.0	28.0	80.0
PolarNet [26]	12.0	32.0	84.0	44.0	56.0	12.0	4.0	8.0	52.0	80.0
PerAct [43]	28.0±4.4	51.2±4.7	84.0±3.6	17.6±2.0	74.0±13.0	16.8±4.7	26.4±3.2	2.4±2.0	52.0±0.0	88.0±4.4
Act3D [12]	51.0	90.0	95.0	47.0	93.0	8.0	12.0	9.0	92.0	94.0
RVT [13]	49.6±3.2	88.0±5.7	91.2±3.0	48.0±5.7	81.6±5.4	36.0±2.5	28.8±3.9	26.4±8.2	72.0±0.0	93.6±4.1
3D Diffuser Actor [11]	**85.6±4.1**	96.0±3.6	97.6±2.0	82.4±2.0	**97.6±3.2**	44.0±4.4	68.3±3.3	47.2±8.5	84.0±4.4	**99.2±1.6**
RVT-2 [14]	66.0±4.5	96.0±0.0	96.0±2.8	**88.0±4.9**	92.0±2.8	35.0±7.1	**80.0±2.8**	69.0±5.9	**100.0±0.0**	99.0±1.7
BridgeVLA (Ours)	73.6±4.6	**99.2±1.8**	**99.2±1.8**	87.2±6.6	96.0±2.8	**60.8±7.7**	76.8±8.7	**81.6±3.6**	87.2±1.8	92.8±3.3

Table 1 Results on RLBench. The "Avg. Rank" column reports the average rank of each method across all 18 tasks, where lower values indicate better overall performance. BridgeVLA achieves the best performance in 10 out of 18 tasks.

The training loss during fine-tuning consists of four components:

$$L = L_{\text{trans}} + L_{\text{rot}} + L_{\text{gripper}} + L_{\text{collision}} \tag{4}$$

Similar to pre-training, L_{trans} is a cross-entropy loss that supervises the heatmap prediction for translational actions. The ground-truth heatmap for each orthographic view is the normalized single-object probability map defined in Eq. 2, where $\widehat{\mathbf{x}}_i$ represents the projected pixel position of the ground-truth end-effector position in the next keyframe. As we discretize the Euler angles for rotation into bins, we also apply cross-entropy loss in L_{rot} to supervise rotation prediction. For gripper action and collision avoidance, we use the binary cross-entropy loss in L_{gripper} and $L_{\text{collision}}$ as supervision. To enhance geometric robustness, random rigid-body transformations are applied jointly to the point cloud and the ground-truth action during training. Additional training details can be found in Appendix A.

4 Experiments

In this section, we perform extensive experiments in both simulation and the real world to evaluate the proposed method. Through the experiments, we aim to answer four questions:

Q1: How effective does BridgeVLA learn 3D robot manipulation compared to state-of-the-art methods?

Q2: Is BridgeVLA capable of learning efficiently from very limited data (*e.g.*, 3 trajectories per task)?

Q3: How robust is BridgeVLA on handling visual disturbances (*e.g.*, distractors, background, and lighting)?

Q4: How does BridgeVLA generalize to novel object-skill combinations and objects from previously unseen categories?

4.1 Simulation Experiments

4.1.1 Experiments on RLBench

Setup. RLBench [17] implements tasks in CoppeliaSim [44] using a Franka Panda robot mounted with a parallel-jaw gripper. The observation contains four RGB-D images captured from four calibrated cameras

	Overall		**Success Rate (%)**					
Models	**Avg. SR (%) ↑**	**Avg. Rank ↓**	All Perturbations	MO-COLOR	RO-COLOR	MO-TEXTURE	RO-TEXTURE	MO-SIZE
R3M-MLP[46]	0.8	5.71	0.6	0.4	0.0	0.0	0.0	1.8
MVP-MLP[47]	1.6	5.0	0.8	1.2	0.0	0.4	0.0	4.44
PerAct[43]	27.9	3.71	7.2	24.0	29.2	28.8	17.71	35.6
RVT[13]	35.4	3.28	6.4	26.0	31.3	44.8	41.1	35.3
RVT-2[14]	56.7	1.92	15.6 ± 0.8	53.0 ± 0.9	54.6 ± 0.6	59.7 ± 0.7	56.7 ± 1.4	60.9 ± 0.9
BridgeVLA (Ours)	**64.0**	**1.07**	**18.7 ± 2.2**	**60.5 ± 1.1**	**63.8 ± 0.1**	**63.5 ± 1.5**	**68.4 ± 3.3**	**69.3 ± 1.0**
Models	RO-SIZE	Light Color	Table Color	Table Texture	Distractor	Background Texture	RLBench	Camera Pose
R3M-MLP[46]	0.0	1.0	1.4	0.2	1.6	1.2	2.0	0.8
MVP-MLP[47]	0.0	1.6	1.6	1.0	3.8	2.2	2.0	2.6
PerAct[43]	29.3	29.1	30.4	23.2	27.1	33.5	39.4	36.3
RVT[13]	40.5	34.0	30.0	45.2	18.8	46.4	53.4	42.2
RVT-2[14]	53.4 ± 1.5	58.0 ± 1.1	62.6 ± 0.9	56.6 ± 0.9	**60.8 ± 0.5**	68.7 ± 1.1	68.8 ± 1.3	64.4 ± 0.5
BridgeVLA (Ours)	**61.7 ± 0.8**	**69.7 ± 1.2**	**75.7 ± 0.9**	**71.3 ± 0.7**	51.8 ± 1.5	**74.8 ± 1.0**	**73.1 ± 0.2**	**73.8 ± 0.3**

Table 2 Results on the COLOSSEUM Benchmark. The table shows the success rates across 14 generalization settings. The "Avg. Rank" column reports the average rank of each method across all perturbations, where lower values indicate better overall performance. Compared to the state-of-the-art baseline, BridgeVLA improves the average success rate by 7.3%.

positioned at the front, left shoulder, right should, and wrist. Following previous works [10–14], we perform experiments on 18 tasks from RLBench. These tasks span 1) non-prehensile manipulation (*e.g.*, *slide block to target*), 2) pick-and-place (*e.g.*, *stack cups*), and 3) high-precision insertion (*e.g.*, *insert peg*). Each task is provided with 100 expert demonstrations. And each demonstration is paired with language instruction and multiple keyframes. Models are evaluated via binary success rates over 25 trials per task, with a maximum of 25 action steps per trial.

Baselines. We compare BridgeVLA with multiple baselines. (1) **Image-BC (CNN)** and **Image-BC (ViT)** [41] are two 2D baseline methods which predict the actions directly from 2D images using CNN and ViT as the backbone, respectively. (2) **C2F-ARM-BC** [29] predicts the next keyframe action in the voxel space with a coarse-to-fine strategy. (3) **PerAct** [10] also operates in the voxel space and predicts the action with a perciever transformer [43]. (4) **HiveFormer** incorporates historical information using a unified multi-modal transformer architecture. (5) **PolarNet** employs PointNext [45] to encode the 3D scene and predicts both heatmaps and offsets for all points to estimate translational actions. (6) **Act3D** [12] predicts the next keyframe action by selecting the point with the highest score from a set of randomly sampled points in the workspace. (7) **3D Diffuser Actor** [11] generates 3D trajectories via a diffusion process conditioned on 3D observation and language instructions. (8) **RVT** [13] uses multi-view transformer to aggregate information from multiple orthographic views of the point cloud observation. (9) And **RVT-2** [14], the current state-of-the-art method, further improves the precision of its prior via a coarse-to-fine strategy.

Results. In total, we evaluate BridgeVLA five times to minimize statistical bias. The results are shown in Table 1. BridgeVLA outperforms all the comparing baseline methods, achieving an average success rate of 88.2% and an average rank of 1.9 across all the 18 tasks, establishing a new state of the art in this benchmark. These results address Q1, demonstrating the effectiveness of BridgeVLA in learning complex 3D manipulation tasks. We highlight that BridgeVLA outperforms the best baseline method by a large margin in *Insert Peg* (88.0% vs 40.0%) and *Sort Shape* (60.8% vs 35.0%). These two tasks demand highly precise alignment between the peg and hole and the block and sorter, respectively. The high success rates of our method showcase its strong capabilities of learning precise manipulation which is highly desirable in many industrial applications. Among the 18 tasks, BridgeVLA performs the worst in *Place Cups*, despite surpassing all the comparing baseline methods. We hypothesize this is because the target keypoints are often occluded in all orthographic projection views, which makes learning and prediction more challenging. In the future, we plan to explore dynamically selecting the projection views for rendering to avoid this problem.

Method	Average	L1	L2	L3	L4
Hiveformer [42]	30.4	60.3 ± 1.5	26.1 ± 1.4	35.1 ± 1.7	0.0 ± 0.0
PolarNet [26]	38.4	77.7 ± 0.9	37.1 ± 1.4	38.5 ± 1.7	0.1 ± 0.2
3D Diffuser Actor [11]	44.0	91.9 ± 0.8	43.4 ± 2.8	37.0 ± 2.2	0.0 ± 0.0
RVT-2 [14]	44.0	89.1 ± 0.8	51.0 ± 2.3	36.0 ± 2.2	0.0 ± 0.0
3D-LOTUS [19]	45.7	**94.3** ± 1.4	49.9 ± 2.2	38.1 ± 1.1	0.3 ± 0.3
3D-LOTUS$_{++}$ [19]	48.0	68.7 ± 0.6	64.5 ± 0.9	41.5 ± 1.8	**17.4** ± 0.4
BridgeVLA (Ours)	**50.0**	91.1 ± 1.1	**65.0** ± 1.3	**43.8** ± 1.2	0.0 ± 0.0

Table 3 Results on GemBench. We show the average success rates on the four evaluation settings of GemBench. BridgeVLA establishes a new state of the art on this benchmark, achieving an average success rate of 50.0%.

4.1.2 Experiments on COLOSSEUM

Setup. To systematically evaluate the generalization capabilities of BridgeVLA, we further evaluate on the COLOSSEUM benchmark [18]. The COLOSSEUM benchmark is an extension to the RLBench benchmark. The model is trained on the data from the original RLBench benchmark but evaluated in environments spanning 12 axes of perturbations. These perturbations, which are unseen during training, encompass changes in object texture, color, and size, backgrounds, lighting, distractors and camera poses. In total, the COLOSSEUM creates 20,371 unique task perturbations instances to comprehensively evaluate the generalization capabilities of the model. Specifically, our evaluation includes three steps: 1) train the model with the original RLBench data without perturbations (100 trajectories per task) on 20 tasks, 2) evaluate each task over 25 trials per perturbation, 3) compute the average success rate of all evaluated tasks for every perturbation. Besides the 12 types of perturbations, we also evaluate on basic variations from the original RLBench (denoted as **RLBench** in Tab. 2), and a more challenging setting which combines all the 12 types of perturbations (denoted as **All Perturbations** in Tab. 2).

Baselines. We compare BridgeVLA with five baseline methods. **R3M-MLP** and **MVP-MLP** are two 2D methods that utilize pre-trained visual encoders to process observation images and an MLP for action prediction. Specifically, R3M-MLP uses R3M [46] that is pre-trained on large-scale egocentric human videos; MVP-MLP uses MVP [47] that is pre-trained on millions of in-the-wild data. Both visual encoders show strong adaptability on various robotics tasks in both simulation and the real world. We also compare with three 3D methods introduced in Sec. 4.1.1, *i.e.*, **PerAct** [10], **RVT** [13], and **RVT-2** [14].

Results. Results are shown in Tab. 2. BridgeVLA outperforms all the comparing baseline methods in terms of average success rate, significantly outperforming the best baseline method by 7.3%. Among all the 14 evaluated perturbations, our method ranks the best among all methods in 13 of them. These results address Q3, showcasing that BridgeVLA possesses strong robustness against visual perturbation. More detailed results can be found in Tab. 5 and 6.

4.1.3 Experiments on GemBench

Setup. To further evaluate the generalization capabilities of BridgeVLA, we perform experiments on the GemBench benchmark. GemBench [19] is a hierarchical generalization benchmark built on the RLBench simulator [17]. Its training set contains 16 tasks (31 variations) covering seven core action primitives—press, pick, push, screw, close, open, and stack/put. The test set consists of 44 tasks (92 variations), categorized into four increasingly challenging settings:

L1 (Novel Placements): L1 consists of the original 16 tasks (31 variations). The object placements are randomized within the workspace. In addition, chromatic distractors are introduced to test the ability to handle additional visual complexity.

L2 (Novel Rigid Objects): L2 involves 15 unseen tasks (28 variations) that require interaction with 8 novel rigid objects using learned primitives. The generalization capabilities are evaluated across two categories: novel object-color compositions and novel object shapes.

L3 (Novel Articulated Objects): L3 consists of 18 unseen tasks (21 variations) that involve interacting with articulated objects. It evaluates the generalization capabilities across three categories: novel action-part compositions, novel object instances, and novel object categories.

L4 (Novel Long-Horizon Tasks): L4 includes 6 complex long-horizon tasks (12 variations) that require combining multiple actions to finish a whole task.

Baselines. In total, we compare with six baseline methods. **3D-LOTUS** [19] processes point cloud inputs through a language-conditioned point cloud transformer architecture [48]. It showcases notable multi-tasking capabilities and high training efficiency. Its enhanced variant, **3D-LOTUS++** [19], integrates the generalization capabilities of large-scale models into 3D-LOTUS with a modular architecture consisting of three components: (1) LLM-based task planning [49], (2) VLM-based object grounding [50, 51], and (3) motion control inherited from 3D-LOTUS. We also compare with four methods introduced in Sec. 4.1.1, *i.e.*, **Hiveformer** [42], **PolarNet** [26], **3D Diffuser Actor** [11], **RVT-2** [14]

Results. Results are shown in Tab. 3 and more detailed results can be found in Appendix B.4. BridgeVLA consistently outperforms all the comparing baseline methods in terms of average success rate across the four evaluation settings. Notably, BridgeVLA achieves state-of-the-art results in both the L2 and L3 settings, demonstrating strong generalization capabilities, addressing Q4. However, similar to most baseline approaches, BridgeVLA exhibits limited performance in the L4 setting, where each task comprises multiple sub-tasks. In the future, we plan to explore leveraging large language models (LLMs) for long-horizon task decomposition and further improve the performance in such setting.

4.2 Real-Robot Experiments

Setup. In this section, we perform real-robot experiments to validate the effectiveness of BridgeVLA in the real world. Our real-robot setup includes a Franka Research 3 robot arm mounted with a parallel-jaw gripper (Fig. 3). A static ZED 2i depth camera is used to provide the colored point cloud observation. In total, we evaluate on 13 tasks (see Tab. 11 for a full list of tasks). These tasks ranges from simple pick-and-place to complex long-horizon tasks, requiring the robot to open a drawer and put items into the drawer. Each task contains 3-9 keyframes (see Fig. 7 and 8 for visualization). For each task, we collect 10 expert trajectories for training.

In total, we design 7 different settings to comprehensively evaluate our model's performance. (1) **Basic**: The model is evaluated in environments that are similar to the training data. (2) **Distractor**: Distractor objects that are visually similar to at least one target object are added to the scene. (3) **Lighting**: The model is tested in a visually distinct lighting condition in which the lights are turned off. (4) **Background**: Three different tablecloths are used to change the background. (5) **Height**: All objects for manipulation are placed on a drawer that is 9.5cm high. (6) **Combination**: We combine objects and skills that are not paired together in the training datasets. That is, while the objects (*e.g.*, red block and green plate) and skill (*e.g.*, place A in B) are seen during training, the instruction that pairs them together is novel (*e.g.*, place the red block in the green plate). In total, we evaluate 13 novel object-skill combinations (Fig. 11 and 12). (7) **Category**: To test whether BridgeVLA is able to transfer the broad knowledge from pre-training to downstream policy learning, we evaluate on manipulating objects from categories that are *unseen* in the robot training data. In total, we test 7 novel objects (Fig. 13). Distractor, Lighting, Background, and Height aim to evaluate the robustness against visual disturbances, while Combination and Category evaluate the generalization capabilities for unseen instructions.

We compare with **RVT-2** [14], the strongest baseline method in the simulation experiments (Sec. 4.1). In addition, we compare with an ablated variant of BridgeVLA, **BridgeVLA w/o Pre-train** which excludes the 2D-heatmap pre-training. This ablation study helps us understand the effectiveness of our proposed pre-training method.

Results. Results are shown in Fig. 3. BridgeVLA outperforms both comparing baseline methods in six out of the seven settings. All the three methods perform well in the Basic setting. However, RVT-2 struggles

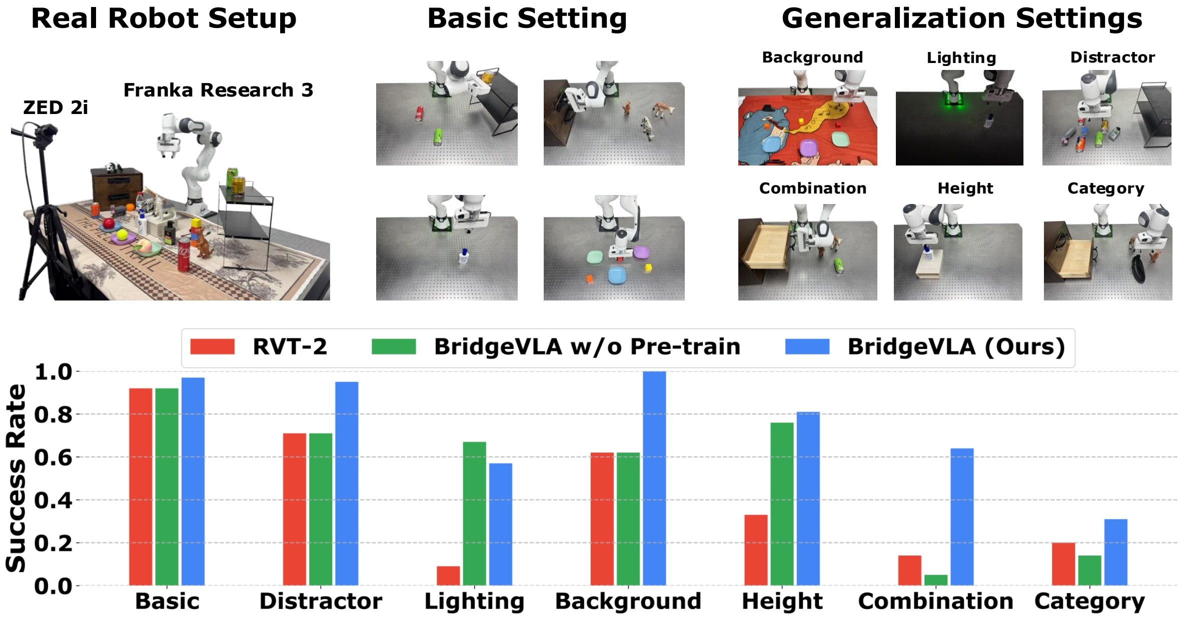


Figure 3 Real-Robot Experiments and Results. We use a Franka Research 3 robot arm and a ZED 2i camera to capture point clouds of the scene. To evaluate the model's performance, we design 7 different settings including one basic setting and six generalization settings. Experimental results show that BridgeVLA outperforms the state-of-the-art baseline method RVT-2 [14] by an average of 32%.

in all the four visual disturbance settings, while BridgeVLA performs better and is able to maintain high performance in Distractor and Background settings. These results addresses Q3, indicating that BridgeVLA is able to handle visual disturbance robustly. To assess the data efficiency of BridgeVLA, we also train the model with only 3 trajectories per task. Remarkably, despite the limited data, BridgeVLA achieves a success rate of 96.8% in Basic, matching the performance achieved with 10 trajectories per task. This result underscores the data efficiency of the proposed method, directly addressing Q2. Detailed per-task results are provided in Appendix C.4.

BridgeVLA w/o Pre-train is not able to generalize well in both language-related generalization settings. Using CLIP [52] as the text encoder, RVT-2 is able to outperform BridgeVLA w/o Pre-train in both settings. BridgeVLA achieves the best performance across the two generalization settings. It significantly surpasses the two baseline methods in Combination, highlighting its ability to understand language semantics. We hypothesize that the 2D-heatmap pre-training equips BridgeVLA with the ability to connect the semantics in language instructions with image observations in the heatmap space. These results address Q4 and highlight the effectiveness and importance of the proposed 2D-heatmap pre-training.

Although our method outperforms baseline methods in the Category setting, its absolute success rate is not high. A common failure mode is that the robot often ignores the target object and moves directly to the destination during pick-and-place manipulation. We believe this relatively low performance is *not* due to BridgeVLA forgetting the knowledge gained from pre-training, as it still predicts heatmaps accurately when provided with samples from the pre-training dataset after fine-tuning (see Fig. 4 and Appendix C.3). Instead, we hypothesize that the reduced performance stems from two factors: 1) The images in the pre-training dataset are mostly captured from third-person views, which differ significantly from the projection images in our robot data; 2) The pre-training task focuses solely on object localization, whereas manipulation involves predicting keypoints that do not correspond to an object. To address these issues, we plan to expand both the scale and diversity of the pre-training dataset and explore more expressive action-decoding methods to better leverage the preserved pre-training knowledge.

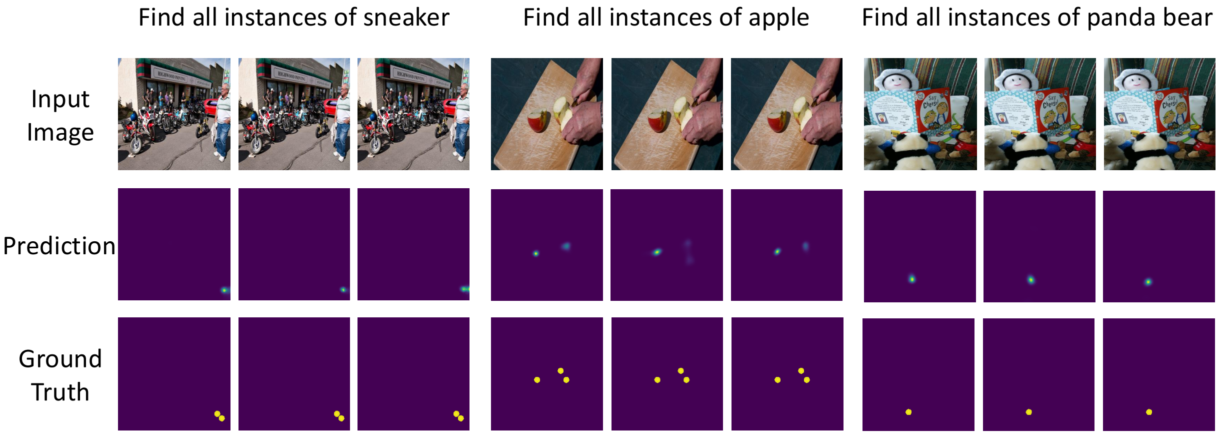


Figure 4 Prediction on Pre-training Data after Fine-tuning. To simulate the multi-view inputs during fine-tuning, we repeat each pre-training image three times and feed them into the fine-tuned model to generate heatmaps. Note that these samples are *not* cherry-picked. Additional samples can be found in Appendix C.3.

5 Conclusions & Future Work

This paper has introduced BridgeVLA, a novel and efficient 3D vision-language-action (VLA) model built on top of a pre-trained vision-language model (VLM) [1]. Keys to our method are that (1) it converts 3D inputs to 2D images to align with the 2D image inputs of the pre-trained VLM; (2) it aligns the input observation and the output action to a unified 2D image space via 2D heatmap prediction; (3) it adopts a scalable pre-training method to equip the VLM with the capability to predict heatmaps before fine-tuning on action prediction. Extensive experiments show that the proposed method is able to learn 3D manipulation efficiently and effectively in both simulation and the real world. In the future, we plan to explore pre-training on more diverse tasks, including semantic segmentation and keypoint detection. We also want to incorporate more expressive action-decoding methods (*e.g.*, diffusion [21]) into the framework to continue improving the policy performance.

References

[1] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726, 2024.

[2] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191, 2024.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923, 2025.

[4] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In Forty-first International Conference on Machine Learning, 2024.

[5] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In 8th Annual Conference on Robot Learning.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. URL https://arxiv. org/abs/2410.24164.

[7] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi 0.5$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054, 2025.

[8] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. arXiv preprint arXiv:2311.01378, 2023.

[9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818, 2023.

[10] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In Conference on Robot Learning, pages 785–799. PMLR, 2023.

[11] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In 8th Annual Conference on Robot Learning, 2024.

[12] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In Conference on Robot Learning, pages 3949–3965. PMLR, 2023.

[13] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning, pages 694–710. PMLR, 2023.

[14] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. In RSS 2024 Workshop: Data Generation for Robotics, 2024.

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

[16] 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.

[17] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters, 5(2):3019–3026, 2020.

[18] Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. arXiv preprint arXiv:2402.08191, 2024.

[19] Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. arXiv preprint arXiv:2410.01345, 2024.

[20] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817, 2022.

[21] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research, 2024.

[22] Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. IEEE Robotics and Automation Letters, 2025.

[23] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. arXiv preprint arXiv:2410.06158, 2024.

[24] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705, 2023.

[25] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In International Conference on Learning Representations, 2024.

[26] Shizhe Chen, Ricardo Garcia Pinel, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. In Conference on Robot Learning, pages 1761–1781. PMLR, 2023.

[27] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place. arXiv preprint arXiv:2311.00926, 2023.

[28] Rujia Yang, Geng Chen, Chuan Wen, and Yang Gao. Fp3: A 3d foundation policy for robotic manipulation. arXiv preprint arXiv:2503.08950, 2025.

[29] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13739–13748, 2022.

[30] 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.

[31] 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.

[32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193, 2023.

[33] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In 2021 IEEE international conference on robotics and automation (ICRA), pages 4613–4619. IEEE, 2021.

[34] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. IEEE Robotics & Automation Magazine, 19(4):72–82, December 2012. https://ompl.kavrakilab.org.

[35] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065), volume 2, pages 995–1001. IEEE, 2000.

[36] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. arXiv preprint arXiv:1404.3785, 2014.

[37] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. arXiv preprint arXiv:2406.10721, 2024.

[38] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In Proceedings of the IEEE/CVF international conference on computer vision, pages 11975–11986, 2023.

[39] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295, 2024.

[40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, pages 402–419. Springer, 2020.

[41] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In Conference on Robot Learning, pages 991–1002. PMLR, 2022.

[42] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In Conference on Robot Learning, pages 175–187. PMLR, 2023.

[43] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. In International Conference on Learning Representations.

[44] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In 2013 IEEE/RSJ international conference on intelligent robots and systems, pages 1321–1326. IEEE, 2013.

[45] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. Advances in neural information processing systems, 35:23192–23204, 2022.

[46] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. arXiv preprint arXiv:2203.12601, 2022.

[47] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. arXiv preprint arXiv:2203.06173, 2022.

[48] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4840–4851, 2024.

[49] AI@Meta. Llama 3 model card. 2024.

[50] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. Advances in Neural Information Processing Systems, 36:72983–73007, 2023.

[51] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. arXiv:2304.02643, 2023.

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PmLR, 2021.

	Pretrain	RLBench Finetune	COLOSSEUM Finetune	GemBench Finetune	Real-robot Finetune
learning rate	5e-5	8e-5	8e-5	8e-5	2e-5
optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
batch size	384	192	192	160	192
warmup steps	400	–	–	–	–

Table 4 Training hyperparameters for BridgeVLA

Appendix

A Training

Detailed training configurations are summarized in Tab. 4. Throughout both pre-training and fine-tuning, we keep the SigLIP vision encoder and language token embeddings frozen.

Computational Resources:

1. Pre-training: 8 NVIDIA A100 GPUs for 3,800 steps (≈2 hours)
2. RLBench fine-tuning: 48 NVIDIA H100 GPUs for 83,000 steps (≈20 hours)
3. COLOSSEUM fine-tuning: 48 NVIDIA H100 GPUs for 83,000 steps (≈20 hours)
4. GemBench fine-tuning: 40 NVIDIA A100 GPUs for 50 epochs (≈2.1 hours)
5. Real-world fine-tuning: 8 NVIDIA A100 GPUs for 300 epochs (≈1.5 hours)

B Simulation Experiments

B.1 Key frame Selection

For all the simulation and real-robot experiments, we adopt the same key frame selection strategy as PerAct [10]. A time step is labeled as a key frame if (i) the robot is stationary, (ii) the gripper state changes, or (iii) the step is the final state of the episode. The robot is considered stationary when the absolute velocities of all joints fall below 0.1 rad/s.

B.2 Data

Following [10, 13, 14], we select 18 tasks from RLBench [17] to evaluate the performance of our method on complex manipulation tasks. These tasks are visualized in Fig. 5.

To assess the generalization capability of BridgeVLA, we also evaluate on the COLOSSEUM benchmark [18] and GemBench [19]. The COLOSSEUM benchmark includes 20 basic tasks and 12 types of perturbations. These perturbations, which are unseen during training, encompass changes in object texture, color, and size, backgrounds, lighting, distractors and camera poses. The benchmark evaluates on all the 12 types of perturbations, a setting with basic variations from the original RLBench, and a more challenging setting which combines all the 12 types of perturbations. We visualize all perturbations except the one from the original RLBench in Fig. 6.

For GemBench, the training set includes 16 tasks (31 variations) spanning seven fundamental action primitives (press, pick, push, screw, close, open, stack/put). The test set includes 44 tasks (92 variations) organized into four increasingly challenging settings. Unlike RLBench and COLOSSEUM, where demo augmentation is used, we train BridgeVLA using only keyframes from each trajectory without performing any demo augmentation in GemBench.

B.3 Detailed Results on COLOSSEUM

For the COLOSSEUM results in Tab.2, we use the results of R3M-MLP [46], MVP-MLP [47], RVT [13], and PerAct [10] from the original COLOSSEUM paper [18]. For RVT-2 [14] and BridgeVLA, we perform our own training and evaluation process. We performed three test repetitions and report the average success rate and variance of BridgeVLA and RVT-2 for each task under different perturbations in Tab.5 and Tab.6, respectively.

B.4 Detailed Results on GemBench

We show per-task success rates on the four settings of GemBench in Tab. 7, 8, 9, 10. The results of baseline methods are sourced from [19]. In total, we evaluate on 5 random seeds to reduce statistical variance. And for every seed, we run 20 trials per task variation.

C Real-Robot Experiments

C.1 Experiment Setup

Fig. 3 illustrates our real-robot setting. The platform comprises a 7-DoF Franka Research 3 manipulator with a parallel-jaw gripper and a ZED 2i stereo camera mounted on a tripod for capturing point clouds of the workspace. We collect expert trajectories with a kinestheic teaching approach. We first move the manipulator to keypoints of an expert trajectory and then play back the keypoints to record the observation and action at each keypoint.

C.2 Generalization Settings

We evaluate on a total of six generalization settings: Distractor, Lighting, Background, Height, Combination, and Category. For Distractor, Lighting, Background, and Height, we visualize these settings in Fig. 10. We visualize the settings of Combination and Category in Fig. 11 and Fig. 12, respectively.

In Distractor, we add distractor objects that are visually similar to at least one target object to the scene. In Lighting, we evaluate the model in a novel lighting condition in which the lights are off. In Background, we use three different tablecloths to change the background. For Height, we elevate all objects for manipulation with a drawer that is about 10cm high. Distractor, Lighting, Background, and Height aim to evaluate the robustness against visual disturbances.

In Combination, we combine objects and skills that are not paired together in the training datasets. That is, while the object for manipulation and the manipulation skill are seen during training, the instruction that pairs them together is novel. The setting of Combination helps us evaluate whether the model is able to generalize across novel object-skill combinations. In Category, we want to evaluate whether BridgeVLA is able to manipulate objects from categories that are *unseen* in the robot training data. In total, we test 7 novel objects.

C.3 Preservation of Object Grounding Capability after Fine-tuning

We observe that even after fine-tuning on robot action data, BridgeVLA retains the object grounding capability learned during pre-training. We visualize its predictions on the pre-training dataset after fine-tuning in Fig. 14. It is important to note that the samples in Fig. 14 are not cherry-picked. BridgeVLA does not forget its pre-training knowledge after 3D action fine-tuning.

C.4 Per-task Success Rate

We showcase per-task success rates of BridgeVLA in the basic setting in Tab. 11. Notably, BridgeVLA achieves exceptionally high success rates even with only 3 trajectories per task, highlighting its superb sample efficiency.

Task Name	Original	All Perturbations	MO-COLOR	RO-COLOR	MO-TEXTURE	RO-TEXTURE	MO-SIZE	RO-SIZE	Light Color	Table Color	Table Texture	Distractor	Background Texture	RLBench	Camera Pose
basketball_in_hoop	100.0±0.0	4.0±3.3	94.7±1.9	96.0±0.0	84.0±5.7	-	100.0±0.0	68.0±0.0	100.0±0.0	100.0±0.0	100.0±0.0	37.3±1.9	100.0±0.0	100.0±0.0	100.0±0.0
close_box	100.0±0.0	72.0±0.0	94.7±1.9	-	-	-	93.3±1.9	-	100.0±0.0	100.0±0.0	98.7±1.9	98.7±1.9	100.0±0.0	97.3±1.9	100.0±0.0
close_laptop_lid	100.0±0.0	11.1±15.7	82.7±3.8	-	-	-	67.9±14.6	-	89.3±8.2	92.0±0.0	97.3±3.8	82.7±6.8	96.0±3.3	100.0±0.0	96.0±0.0
empty_dishwasher	0.0±0.0	0.0±0.0	1.3±1.9	1.3±1.9	-	1.3±1.9	4.0±3.3	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	1.3±1.9	1.3±1.9	0.0±0.0
get_ice_from_fridge	94.7±1.9	5.3±1.9	86.7±1.9	90.7±7.5	90.7±5.0	-	84.0±3.3	73.3±1.9	96.0±3.3	98.7±1.9	89.3±7.5	56.0±8.6	94.7±1.9	96.0±3.3	98.7±1.9
hockey	57.3±5.0	9.3±3.8	44.0±6.5	50.7±8.2	-	50.7±13.2	46.7±8.2	65.3±5.0	45.3±1.9	64.0±8.6	53.3±1.9	20.0±3.3	56.0±5.7	49.3±5.0	50.7±5.0
insert_onto_square_peg	93.3±3.8	23.3±2.4	52.0±3.3	94.7±1.9	-	76.0±8.6	85.3±3.8	70.7±3.8	84.0±0.0	88.0±3.3	88.0±3.3	44.0±11.8	86.7±1.9	77.3±5.0	96.0±0.0
meat_on_grill	96.0±0.0	9.3±1.9	32.0±0.0	88.0±5.7	-	-	100.0±0.0	-	100.0±0.0	92.0±6.5	90.7±1.9	98.7±1.9	97.3±1.9	100.0±0.0	100.0±0.0
move_hanger	37.3±3.8	2.7±3.8	26.7±3.8	46.7±3.8	-	-	-	-	52.0±0.0	84.0±0.0	52.0±5.7	52.0±5.7	33.3±5.0	42.7±1.9	24.0±0.0
open_drawer	96.0±0.0	60.0±3.3	97.3±1.9	-	-	-	90.7±1.9	-	88.0±3.3	93.3±1.9	100.0±0.0	90.7±1.9	100.0±0.0	94.7±1.9	96.0±0.0
place_wine_at_rack_location	88.0±5.7	17.3±13.6	82.7±5.0	89.3±7.5	-	92.0±6.5	93.3±3.8	90.7±3.8	90.7±5.0	97.3±1.9	88.0±3.3	74.7±3.8	90.7±6.8	92.0±3.3	92.0±8.6
put_money_in_safe	94.7±1.9	6.7±5.0	78.7±1.9	74.7±1.9	81.3±6.8	89.3±5.0	92.0±3.3	-	37.3±12.4	84.0±3.3	84.0±3.3	84.0±3.3	89.3±1.9	86.7±8.2	86.7±1.9
reach_and_drag	100.0±0.0	0.0±0.0	89.3±3.8	96.0±0.0	94.7±5.0	84.0±5.7	94.7±1.9	38.7±5.0	92.0±3.3	88.0±5.7	78.7±3.8	28.0±8.6	100.0±0.0	100.0±0.0	94.7±3.8
scoop_with_spatula	96.0±3.3	6.7±1.9	94.7±1.9	93.3±1.9	85.3±3.8	85.3±3.8	78.7±3.8	86.7±5.0	90.7±1.9	88.0±6.5	77.3±1.9	20.0±5.7	90.7±6.8	89.3±1.9	93.3±1.9
setup_chess	10.7±1.9	0.0±0.0	1.3±1.9	8.0±0.0	8.0±3.3	-	13.3±1.9	-	12.0±5.7	21.3±8.2	13.3±3.8	5.3±1.9	20.0±5.7	16.0±5.7	4.0±3.3
slide_block_to_target	100.0±0.0	24.0±3.3	74.7±1.9	-	92.0±3.3	-	-	-	100.0±0.0	100.0±0.0	98.7±1.9	84.0±9.8	100.0±0.0	100.0±0.0	100.0±0.0
stack_cups	58.7±3.8	29.3±1.9	66.7±1.9	-	50.7±1.9	-	44.0±3.3	-	62.7±1.9	64.0±3.3	65.3±8.2	26.7±7.5	73.3±8.2	64.0±14.2	72.0±8.6
straighten_rope	61.3±6.8	8.0±5.7	16.0±5.7	-	48.0±3.3	-	-	-	61.3±9.4	65.3±1.9	54.7±8.2	37.3±5.0	70.7±8.2	66.7±7.5	72.0±6.5
turn_oven_on	93.3±1.9	85.3±3.8	94.7±3.8	-	-	-	90.7±1.9	-	93.3±3.8	94.7±7.5	96.0±3.3	96.0±3.3	96.0±0.0	88.0±3.3	100.0±0.0
wipe_desk	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	-	0.0±0.0	-	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
Task Mean	73.9±0.7	18.7±2.2	60.5±1.1	63.8±0.1	63.5±1.5	68.4±3.3	69.3±1.0	61.7±0.8	69.7±1.2	75.7±0.9	71.3±0.7	51.8±1.5	74.8±1.0	73.1±0.2	73.8±0.3

Table 5 Success Rates of BridgeVLA under Different Perturbations of COLOSSEUM.

Task Name	Original	All Perturbations	MO-COLOR	RO-COLOR	MO-TEXTURE	RO-TEXTURE	MO-SIZE	RO-SIZE	Light Color	Table Color	Table Texture	Distractor	Background Texture	RLBench	Camera Pose
basketball_in_hoop	100.0±0.0	10.0±2.0	99.0±1.7	94.0±2.0	97.0±1.7	-	100.0±0.0	86.0±3.5	95.0±1.7	94.0±2.0	84.0±6.3	89.0±3.3	100.0±0.0	99.0±1.7	100.0±0.0
close_box	93.0±4.4	36.0±8.5	70.0±6.6	-	-	-	86.0±3.5	-	99.0±1.7	97.0±1.7	91.0±4.4	93.0±3.3	97.0±1.7	94.0±2.0	99.0±1.7
close_laptop_lid	86.0±4.5	40.0±0.0	89.0±3.3	-	-	-	62.0±2.0	-	84.0±4.0	92.0±0.0	96.0±2.8	89.0±5.2	99.0±1.7	87.0±3.3	92.0±0.0
empty_dishwasher	0.0±0.0	0.0±0.0	1.0±1.7	0.0±0.0	-	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
get_ice_from_fridge	95.0±1.7	11.0±4.4	88.0±5.7	77.0±5.2	89.0±1.7	-	78.0±3.5	79.0±3.3	83.0±5.9	89.0±1.7	70.0±4.5	86.0±4.5	81.0±5.2	96.0±2.8	96.0±2.8
hockey	19.0±4.4	0.0±0.0	26.0±4.5	30.0±4.5	-	40.0±4.9	24.0±2.8	13.0±3.3	12.0±8.5	15.0±3.3	9.0±3.3	10.0±6.0	14.0±2.0	17.0±3.3	19.0±3.3
insert_onto_square_peg	31.0±3.3	0.0±0.0	13.0±1.7	35.0±9.5	-	32.0±2.8	33.3±8.6	21.0±1.7	30.0±2.0	9.0±1.7	4.0±4.9	9.0±3.3	35.0±3.3	35.0±1.7	23.0±4.4
meat_on_grill	100.0±0.0	89.0±1.7	100.0±0.0	100.0±0.0	-	-	100.0±0.0	-	99.0±1.7	98.0±2.0	100.0±0.0	99.0±1.7	100.0±0.0	100.0±0.0	100.0±0.0
move_hanger	91.0±5.2	0.0±0.0	61.0±4.4	83.0±18.4	-	-	-	-	55.0±5.9	69.0±5.9	29.0±5.2	92.0±2.8	94.0±2.0	87.0±4.4	22.0±2.0
open_drawer	99.0±1.7	25.0±4.4	63.0±4.4	-	-	-	92.0±0.0	-	88.0±0.0	92.0±0.0	99.0±1.7	86.0±8.2	100.0±0.0	95.0±1.7	95.0±1.7
place_wine_at_rack_location	96.0±4.9	28.0±6.3	74.0±4.5	98.0±2.0	-	93.0±5.2	87.0±3.3	90.0±6.6	81.0±7.1	87.0±4.4	95.0±6.6	83.0±3.3	89.0±5.9	96.0±2.8	91.0±5.2
put_money_in_safe	77.0±4.4	9.0±1.7	45.0±3.3	22.0±3.5	55.0±6.6	73.0±3.3	69.0±1.7	-	56.0±2.8	70.0±4.5	72.0±6.3	82.0±6.6	79.0±3.3	77.0±8.7	62.0±6.0
reach_and_drag	86.0±6.6	0.0±0.0	72.0±5.7	80.0±5.7	60.0±6.9	67.0±5.9	87.0±6.6	55.0±4.4	68.0±2.8	76.0±2.8	71.0±5.2	61.0±6.6	88.0±2.8	86.0±3.5	81.0±5.9
scoop_with_spatula	89.0±5.2	2.0±3.5	75.0±4.4	87.0±3.3	84.0±4.9	92.0±7.5	94.0±4.5	83.0±5.9	54.0±2.0	79.0±5.2	74.0±6.0	83.0±5.9	92.0±2.8	91.0±1.7	89.0±4.4
setup_chess	3.0±1.7	0.0±0.0	0.0±0.0	4.0±2.8	4.0±4.0	-	17.0±7.1	-	7.0±5.2	7.0±3.3	9.0±7.1	14.0±4.5	14.0±3.5	16.0±8.9	9.0±3.3
slide_block_to_target	100.0±0.0	11.0±4.4	45.0±1.7	-	97.0±1.7	-	-	-	84.0±4.9	96.0±0.0	83.0±5.2	82.0±8.7	100.0±0.0	100.0±0.0	100.0±0.0
stack_cups	35.0±5.2	0.0±0.0	47.0±5.9	-	45.0±5.9	-	23.0±4.4	-	18.0±2.0	16.0±4.0	13.0±9.5	19.0±7.7	24.0±2.8	43.0±9.1	40.0±2.8
straighten_rope	66.0±11.5	0.0±0.0	25.0±3.3	-	66.0±10.0	-	-	-	53.0±1.7	68.0±2.8	39.0±11.4	42.0±7.2	72.0±8.5	69.0±6.6	75.0±4.4
turn_oven_on	91.0±4.4	50.0±10.8	68.0±4.9	-	-	-	83.0±1.7	-	95.0±3.3	97.0±1.7	95.0±3.3	96.0±0.0	96.0±4.9	89.0±7.1	96.0±2.8
wipe_desk	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	-	0.0±0.0	-	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
Task Mean	67.8±1.5	15.6±0.8	53.0±0.9	54.6±0.6	59.7±0.7	56.7±1.4	60.9±0.9	53.4±1.5	58.0±1.1	62.6±0.9	56.6±0.9	60.8±0.5	68.7±1.1	68.8±1.3	64.4±0.5

Table 6 Success Rates of RVT-2 under Different Perturbations of COLOSSEUM.

Method	Avg.	Close Fridge+0	Close Jar+15	Close Jar+16	CloseLaptop Lid+0	Close Microwave+0	LightBulb In+17	LightBulb In+19	Open Box+0	Open Door+0	Open Drawer+0
Hiveformer [42]	$60.3_{\pm 1.5}$	$96_{\pm 4.2}$	$64_{\pm 13.9}$	$92_{\pm 2.7}$	$90_{\pm 3.5}$	$88_{\pm 7.6}$	$12_{\pm 4.5}$	$13_{\pm 6.7}$	$4_{\pm 4.2}$	$53_{\pm 15.2}$	$15_{\pm 12.2}$
PolarNet [26]	$77.6_{\pm 0.9}$	$99_{\pm 2.2}$	$99_{\pm 2.2}$	$99_{\pm 2.2}$	$95_{\pm 3.5}$	$98_{\pm 2.7}$	$72_{\pm 12.5}$	$71_{\pm 6.5}$	$32_{\pm 11.5}$	$69_{\pm 8.9}$	$61_{\pm 12.4}$
3D diffuser actor [11]	$91.9_{\pm 0.8}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{99}_{\pm 2.2}$	$\mathbf{100}_{\pm 0.0}$	$85_{\pm 5.0}$	$88_{\pm 2.7}$	$11_{\pm 2.2}$	$96_{\pm 4.2}$	$82_{\pm 9.1}$
RVT-2 [14]	$89.0_{\pm 0.8}$	$77_{\pm 11.0}$	$97_{\pm 4.5}$	$98_{\pm 2.7}$	$77_{\pm 13.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{93}_{\pm 5.7}$	$\mathbf{91}_{\pm 8.2}$	$7_{\pm 4.5}$	$\mathbf{98}_{\pm 4.5}$	$\mathbf{93}_{\pm 5.7}$
3D-LOTUS [19]	$\mathbf{94.3}_{\pm 3.5}$	$96_{\pm 3.7}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$98_{\pm 2.5}$	$98_{\pm 4.0}$	$84_{\pm 7.4}$	$85_{\pm 9.5}$	$\mathbf{99}_{\pm 2.0}$	$77_{\pm 2.5}$	$83_{\pm 8.7}$
3D-LOTUS++ [19]	$68.7_{\pm 0.6}$	$95_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$99_{\pm 2.0}$	$28_{\pm 2.5}$	$87_{\pm 5.1}$	$55_{\pm 10.5}$	$45_{\pm 8.9}$	$55_{\pm 8.9}$	$79_{\pm 9.7}$	$68_{\pm 12.5}$
BridgeVLA (Ours)	$91.1_{\pm 1.1}$	$99_{\pm 2.0}$	$98_{\pm 4.0}$	$\mathbf{100}_{\pm 0.0}$	$97_{\pm 2.5}$	$85_{\pm 5.5}$	$90_{\pm 5.5}$	$87_{\pm 7.5}$	$76_{\pm 10.2}$	$70_{\pm 12.3}$	$86_{\pm 5.8}$

Method	Open Drawer+2	Pick& Lift+0	Pick& Lift+2	Pick& Lift+7	PickUp Cup+8	PickUp Cup+9	PickUp Cup+11	Push Button+0	Push Button+3	Push Button+4	PutIn Cupboard+0
Hiveformer [42]	$59_{\pm 7.4}$	$86_{\pm 4.2}$	$92_{\pm 6.7}$	$93_{\pm 2.7}$	$83_{\pm 7.6}$	$69_{\pm 12.9}$	$61_{\pm 19.8}$	$84_{\pm 11.9}$	$68_{\pm 6.7}$	$87_{\pm 7.6}$	$34_{\pm 8.2}$
PolarNet [26]	$90_{\pm 7.1}$	$92_{\pm 9.1}$	$84_{\pm 7.4}$	$88_{\pm 5.7}$	$82_{\pm 7.6}$	$79_{\pm 4.2}$	$72_{\pm 10.4}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$99_{\pm 2.2}$	$52_{\pm 7.6}$
3D diffuser actor [11]	$97_{\pm 4.5}$	$\mathbf{99}_{\pm 2.2}$	$99_{\pm 2.2}$	$99_{\pm 2.2}$	$96_{\pm 2.2}$	$97_{\pm 4.5}$	$98_{\pm 2.7}$	$98_{\pm 2.7}$	$96_{\pm 4.2}$	$98_{\pm 2.7}$	$85_{\pm 5.0}$
RVT-2 [14]	$94_{\pm 4.2}$	$\mathbf{99}_{\pm 2.2}$	$98_{\pm 2.7}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{99}_{\pm 2.2}$	$\mathbf{99}_{\pm 2.2}$	$\mathbf{99}_{\pm 2.2}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$88_{\pm 8.4}$
3D-LOTUS [19]	$93_{\pm 6.0}$	$\mathbf{99}_{\pm 2.0}$	$\mathbf{100}_{\pm 0.0}$	$99_{\pm 2.0}$	$97_{\pm 4.0}$	$96_{\pm 3.7}$	$94_{\pm 4.9}$	$99_{\pm 2.0}$	$99_{\pm 2.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{89}_{\pm 5.8}$
3D-LOTUS++ [19]	$75_{\pm 4.5}$	$97_{\pm 6.0}$	$94_{\pm 3.7}$	$93_{\pm 5.1}$	$86_{\pm 8.0}$	$88_{\pm 6.8}$	$91_{\pm 4.9}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$1_{\pm 2.0}$
BridgeVLA(Ours)	$\mathbf{99}_{\pm 2.0}$	$\mathbf{99}_{\pm 2.0}$	$\mathbf{100}_{\pm 0.0}$	$98_{\pm 2.5}$	$96_{\pm 2.0}$	$94_{\pm 3.7}$	$\mathbf{99}_{\pm 2.0}$	$\mathbf{100}_{\pm 0.0}$	$98_{\pm 4.0}$	$98_{\pm 4.0}$	$74_{\pm 6.6}$

Method	PutIn Cupboard+3	PutMoney InSafe+0	PutMoney InSafe+1	Reach& Drag+14	Reach& Drag+18	Slide Block+0	Slide Block+1	Stack Blocks+30	Stack Blocks+36	Stack Blocks+39
Hiveformer [42]	$74_{\pm 6.5}$	$85_{\pm 3.5}$	$88_{\pm 2.7}$	$37_{\pm 5.7}$	$32_{\pm 7.6}$	$99_{\pm 2.2}$	$91_{\pm 12.4}$	$6_{\pm 5.5}$	$7_{\pm 4.5}$	$6_{\pm 4.2}$
PolarNet [26]	$\mathbf{88}_{\pm 4.5}$	$93_{\pm 4.5}$	$95_{\pm 5.0}$	$99_{\pm 2.2}$	$99_{\pm 2.2}$	$\mathbf{100}_{\pm 0.0}$	$0_{\pm 0.0}$	$34_{\pm 10.8}$	$30_{\pm 9.4}$	$36_{\pm 12.9}$
3D diffuser actor [11]	$82_{\pm 11.5}$	$\mathbf{95}_{\pm 5.0}$	$98_{\pm 2.7}$	$\mathbf{100}_{\pm 0.0}$	$99_{\pm 2.2}$	$\mathbf{100}_{\pm 0.0}$	$89_{\pm 4.2}$	$88_{\pm 7.6}$	$85_{\pm 6.1}$	$89_{\pm 5.5}$
RVT-2 [14]	$80_{\pm 6.1}$	$93_{\pm 8.4}$	$96_{\pm 8.5}$	$85_{\pm 10.0}$	$94_{\pm 2.2}$	$\mathbf{100}_{\pm 0.0}$	$37_{\pm 6.7}$	$88_{\pm 5.7}$	$\mathbf{93}_{\pm 2.7}$	$88_{\pm 11.5}$
3D-LOTUS [19]	$72_{\pm 11.2}$	$94_{\pm 3.7}$	$\mathbf{99}_{\pm 2.0}$	$99_{\pm 2.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{94}_{\pm 5.8}$	$91_{\pm 6.6}$	$\mathbf{90}_{\pm 4.5}$
3D-LOTUS++ [19]	$2_{\pm 2.5}$	$22_{\pm 6.8}$	$16_{\pm 4.9}$	$94_{\pm 3.7}$	$62_{\pm 8.7}$	$\mathbf{100}_{\pm 0.0}$	$65_{\pm 5.5}$	$86_{\pm 5.8}$	$20_{\pm 4.5}$	$28_{\pm 13.6}$
BridgeVLA (Ours)	$84_{\pm 6.6}$	$79_{\pm 9.7}$	$86_{\pm 3.7}$	$96_{\pm 5.8}$	$97_{\pm 4.0}$	$\mathbf{100}_{\pm 0.0}$	$90_{\pm 5.5}$	$77_{\pm 8.1}$	$87_{\pm 4.0}$	$85_{\pm 7.8}$

Table 7 Per-task Success Rate on GemBench Level 1.

Method	Avg.	Push Button+13	Push Button+15	Push Button+17	Pick& Lift+14	Pick& Lift+16	Pick& Lift+18	PickUp Cup+10	PickUp Cup+12	PickUp Cup+13
Hiveformer	$26.1_{\pm 1.4}$	$97_{\pm 2.7}$	$85_{\pm 10.0}$	$88_{\pm 2.7}$	$21_{\pm 6.5}$	$9_{\pm 4.2}$	$8_{\pm 6.7}$	$30_{\pm 7.1}$	$22_{\pm 13.5}$	$26_{\pm 10.6}$
PolarNet	$37.1_{\pm 1.4}$	$100_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$85_{\pm 7.9}$	$3_{\pm 4.5}$	$1_{\pm 2.2}$	$0_{\pm 0.0}$	$48_{\pm 11.0}$	$46_{\pm 8.9}$	$16_{\pm 6.5}$
3D diffuser actor	$43.4_{\pm 2.8}$	$87_{\pm 13.0}$	$81_{\pm 6.5}$	$60_{\pm 9.4}$	$9_{\pm 4.2}$	$18_{\pm 9.1}$	$0_{\pm 0.0}$	$84_{\pm 5.5}$	$60_{\pm 11.7}$	$62_{\pm 13.0}$
RVT-2	$51.0_{\pm 2.3}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$47_{\pm 7.6}$	$29_{\pm 9.6}$	$8_{\pm 4.5}$	$81_{\pm 8.2}$	$59_{\pm 9.6}$	$72_{\pm 9.7}$
3D-LOTUS	$49.9_{\pm 2.2}$	$99_{\pm 2.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$3_{\pm 2.5}$	$18_{\pm 8.7}$	$33_{\pm 9.3}$	$89_{\pm 3.7}$	$78_{\pm 8.7}$	$57_{\pm 7.5}$
3D-LOTUS++	$64.5_{\pm 0.9}$	$99_{\pm 2.0}$	$\mathbf{100}_{\pm 0.0}$	$99_{\pm 2.0}$	$\mathbf{94}_{\pm 3.7}$	$\mathbf{96}_{\pm 3.7}$	$\mathbf{95}_{\pm 3.2}$	$79_{\pm 4.9}$	$89_{\pm 9.7}$	$84_{\pm 10.2}$
BridgeVLA (Ours)	$\mathbf{65.0}_{\pm 1.3}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$\mathbf{100}_{\pm 0.0}$	$74_{\pm 9.7}$	$89_{\pm 4.9}$	$0_{\pm 0.0}$	$\mathbf{91}_{\pm 3.7}$	$\mathbf{90}_{\pm 3.2}$	$\mathbf{90}_{\pm 6.3}$

Method	Stack Blocks+24	Stack Blocks+27	Stack Blocks+33	Slide Block+2	Slide Block+3	Close Jar+3	Close Jar+4	LightBulb In+1	LightBulb In+2	Lamp On+0
Hiveformer	$0_{\pm 0.0}$	$4_{\pm 4.2}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$	$4_{\pm 4.2}$	$0_{\pm 0.0}$	$7_{\pm 4.5}$
PolarNet	$1_{\pm 2.2}$	$2_{\pm 2.7}$	$6_{\pm 8.2}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$	$20_{\pm 10.6}$	$82_{\pm 5.7}$	$22_{\pm 11.5}$	$17_{\pm 8.4}$	$\mathbf{14}_{\pm 10.8}$
3D diffuser actor	$\mathbf{66}_{\pm 13.9}$	$82_{\pm 2.7}$	$50_{\pm 14.6}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$	$23_{\pm 16.8}$	$82_{\pm 5.7}$	$51_{\pm 17.8}$	$60_{\pm 10.0}$	$7_{\pm 7.6}$
RVT-2	$18_{\pm 4.5}$	$56_{\pm 16.7}$	$45_{\pm 13.7}$	$0_{\pm 0.0}$	$1_{\pm 2.2}$	$7_{\pm 7.6}$	$77_{\pm 5.7}$	$\mathbf{68}_{\pm 14.4}$	$6_{\pm 6.5}$	$0_{\pm 0.0}$
3D-LOTUS	$13_{\pm 8.1}$	$40_{\pm 9.5}$	$69_{\pm 5.8}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$	$71_{\pm 5.8}$	$90_{\pm 4.5}$	$24_{\pm 4.9}$	$41_{\pm 8.6}$	$0_{\pm 0.0}$
3D-LOTUS++	$22_{\pm 9.3}$	$\mathbf{83}_{\pm 7.5}$	$59_{\pm 3.7}$	$\mathbf{27}_{\pm 9.8}$	$\mathbf{5}_{\pm 3.2}$	$\mathbf{98}_{\pm 2.5}$	$\mathbf{96}_{\pm 3.7}$	$56_{\pm 9.7}$	$43_{\pm 7.5}$	$2_{\pm 2.0}$
BridgeVLA (Ours)	$61_{\pm 10.7}$	$51_{\pm 13.2}$	$\mathbf{79}_{\pm 8.6}$	$12_{\pm 9.3}$	$3_{\pm 4.0}$	$66_{\pm 6.6}$	$88_{\pm 4.0}$	$66_{\pm 8.6}$	$\mathbf{74}_{\pm 5.8}$	$7_{\pm 4.0}$

Method	Reach& Drag+5	Reach& Drag+7	PutCube InSafe+0	Pick&Lift Cylinder+0	Pick&Lift Star+0	Pick&Lift Moon+0	Pick&Lift Toy+0	PutIn Cupboard+7	PutIn Cupboard+8
Hiveformer	$1_{\pm 2.2}$	$0_{\pm 0.0}$	$4_{\pm 2.2}$	$78_{\pm 5.7}$	$73_{\pm 7.6}$	$88_{\pm 2.7}$	$87_{\pm 4.5}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$
PolarNet	$61_{\pm 8.2}$	$10_{\pm 6.1}$	$\mathbf{40}_{\pm 14.1}$	$93_{\pm 6.7}$	$88_{\pm 8.4}$	$93_{\pm 6.7}$	$90_{\pm 3.5}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$
3D diffuser actor	$0_{\pm 0.0}$	$64_{\pm 6.5}$	$3_{\pm 2.7}$	$\mathbf{99}_{\pm 2.2}$	$43_{\pm 17.9}$	$91_{\pm 9.6}$	$30_{\pm 9.4}$	$0_{\pm 0.0}$	$\mathbf{3}_{\pm 4.5}$
RVT-2	$91_{\pm 2.2}$	$89_{\pm 6.5}$	$6_{\pm 5.5}$	$98_{\pm 2.7}$	$98_{\pm 4.5}$	$94_{\pm 4.2}$	$78_{\pm 8.4}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$
3D-LOTUS	$\mathbf{95}_{\pm 4.5}$	$18_{\pm 10.8}$	$25_{\pm 5.5}$	$88_{\pm 8.7}$	$69_{\pm 6.6}$	$80_{\pm 8.4}$	$\mathbf{96}_{\pm 3.7}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$
3D-LOTUS++	$94_{\pm 2.0}$	$64_{\pm 12.4}$	$37_{\pm 5.1}$	$91_{\pm 2.0}$	$94_{\pm 3.7}$	$29_{\pm 6.6}$	$71_{\pm 2.0}$	$\mathbf{1}_{\pm 2.0}$	$0_{\pm 0.0}$
BridgeVLA (Ours)	$94_{\pm 3.7}$	$\mathbf{96}_{\pm 3.7}$	$3_{\pm 2.5}$	$98_{\pm 2.5}$	$\mathbf{99}_{\pm 2.0}$	$\mathbf{95}_{\pm 3.2}$	$93_{\pm 5.1}$	$0_{\pm 0.0}$	$0_{\pm 0.0}$

Table 8 Per-task Success Rate on GemBench Level 2.

Method	Avg.	Close Door+0	Close Box+0	Close Fridge2+0	CloseLaptop Lid2+0	Close Microwave2+0	Open Door2+0	Open Box2+0
Hiveformer	35.1$_{\pm1.7}$	0$_{\pm0.0}$	1$_{\pm2.2}$	34$_{\pm9.6}$	52$_{\pm9.1}$	15$_{\pm7.1}$	32$_{\pm11.5}$	5$_{\pm3.5}$
PolarNet	38.5$_{\pm1.7}$	0$_{\pm0.0}$	0$_{\pm0.0}$	78$_{\pm5.7}$	26$_{\pm8.2}$	74$_{\pm6.5}$	33$_{\pm6.7}$	23$_{\pm8.4}$
3D diffuser actor	37.0$_{\pm2.2}$	0$_{\pm0.0}$	0$_{\pm0.0}$	**97**$_{\pm2.7}$	23$_{\pm6.7}$	88$_{\pm7.6}$	**86**$_{\pm7.4}$	67$_{\pm9.8}$
RVT-2	36.0$_{\pm2.2}$	1$_{\pm2.2}$	2$_{\pm2.7}$	72$_{\pm6.7}$	42$_{\pm14.0}$	71$_{\pm8.9}$	79$_{\pm6.5}$	5$_{\pm6.1}$
3D-LOTUS	38.1$_{\pm1.1}$	0$_{\pm0.0}$	**58**$_{\pm8.1}$	36$_{\pm9.7}$	54$_{\pm10.7}$	85$_{\pm7.1}$	42$_{\pm6.8}$	11$_{\pm6.6}$
3D-LOTUS++	41.5$_{\pm1.8}$	**1**$_{\pm2.0}$	29$_{\pm8.6}$	93$_{\pm2.5}$	50$_{\pm9.5}$	**99**$_{\pm2.0}$	52$_{\pm10.3}$	16$_{\pm8.0}$
BridgeVLA (Ours)	**43.8**$_{\pm1.2}$	0$_{\pm0.0}$	1$_{\pm2.0}$	95$_{\pm5.5}$	**77**$_{\pm4.0}$	54$_{\pm10.2}$	68$_{\pm10.8}$	**74**$_{\pm4.9}$

Method	Open Drawer2+0	Open Drawer3+0	OpenDrawer Long+0	OpenDrawer Long+1	OpenDrawer Long+2	OpenDrawer Long+3	Toilet SeatUp+0	Open Fridge+0
Hiveformer	59$_{\pm11.9}$	39$_{\pm11.9}$	78$_{\pm8.4}$	82$_{\pm4.5}$	49$_{\pm4.2}$	57$_{\pm11.5}$	6$_{\pm4.2}$	0$_{\pm0.0}$
PolarNet	**91**$_{\pm4.2}$	29$_{\pm8.2}$	84$_{\pm11.9}$	**88**$_{\pm5.7}$	**63**$_{\pm8.4}$	37$_{\pm7.6}$	2$_{\pm2.7}$	4$_{\pm2.2}$
3D diffuser actor	19$_{\pm8.2}$	1$_{\pm2.2}$	15$_{\pm5.0}$	35$_{\pm13.7}$	26$_{\pm9.6}$	79$_{\pm12.9}$	0$_{\pm0.0}$	**7**$_{\pm5.7}$
RVT-2	81$_{\pm11.9}$	0$_{\pm0.0}$	**84**$_{\pm8.2}$	39$_{\pm10.8}$	11$_{\pm8.9}$	75$_{\pm6.1}$	7$_{\pm5.7}$	0$_{\pm0.0}$
3D-LOTUS	90$_{\pm3.2}$	22$_{\pm8.1}$	56$_{\pm13.9}$	33$_{\pm11.2}$	17$_{\pm8.1}$	75$_{\pm6.3}$	0$_{\pm0.0}$	4$_{\pm5.8}$
3D-LOTUS++	70$_{\pm5.5}$	41$_{\pm4.9}$	72$_{\pm4.0}$	52$_{\pm10.8}$	23$_{\pm8.1}$	78$_{\pm5.1}$	**8**$_{\pm5.1}$	0$_{\pm0.0}$
BridgeVLA (Ours)	65$_{\pm6.3}$	**87**$_{\pm6.0}$	59$_{\pm8.6}$	34$_{\pm8.0}$	18$_{\pm10.3}$	**85**$_{\pm8.4}$	6$_{\pm5.8}$	**7**$_{\pm2.5}$

Method	OpenLaptop Lid+0	Open Microwave+0	PutMoney InSafe+2	Open Drawer+1	Close Drawer+0	Close Grill+0
Hiveformer	**100**$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	83$_{\pm5.7}$	44$_{\pm10.8}$
PolarNet	**100**$_{\pm0.0}$	0$_{\pm0.0}$	1$_{\pm2.2}$	4$_{\pm4.2}$	29$_{\pm11.9}$	42$_{\pm11.5}$
3D diffuser actor	**100**$_{\pm0.0}$	0$_{\pm0.0}$	2$_{\pm4.5}$	0$_{\pm0.0}$	66$_{\pm7.4}$	**65**$_{\pm13.7}$
RVT-2	93$_{\pm5.7}$	0$_{\pm0.0}$	0$_{\pm0.0}$	**6**$_{\pm2.2}$	78$_{\pm8.4}$	9$_{\pm4.2}$
3D-LOTUS	**100**$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	**87**$_{\pm8.1}$	29$_{\pm6.6}$
3D-LOTUS++	86$_{\pm6.6}$	0$_{\pm0.0}$	**13**$_{\pm8.1}$	0$_{\pm0.0}$	69$_{\pm5.8}$	19$_{\pm13.9}$
BridgeVLA (Ours)	95$_{\pm0.0}$	0$_{\pm0.0}$	2$_{\pm2.5}$	0$_{\pm0.0}$	58$_{\pm12.9}$	35$_{\pm12.3}$

Table 9 Per-task Success Rate on GemBench Level 3.

Method	Avg.	Push Buttons4+1	Push Buttons4+2	Push Buttons4+3	TakeShoes OutOfBox+0	PutItems InDrawer+0	PutItems InDrawer+2
Hiveformer	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
PolarNet	0.1$_{\pm0.2}$	1$_{\pm2.2}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
3D diffuser actor	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
RVT-2	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
3D-LOTUS	0.3$_{\pm0.3}$	3$_{\pm4.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
3D-LOTUS++	**17.4**$_{\pm0.4}$	**76**$_{\pm7.4}$	**49**$_{\pm8.6}$	**37**$_{\pm8.1}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
BridgeVLA (Ours)	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$

Method	PutItems InDrawer+4	Tower4+1	Tower4+3	Stack Cups+0	Stack Cups+3	PutAllGroceries InCupboard+0
Hiveformer	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
PolarNet	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
3D diffuser actor	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
RVT-2	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
3D-LOTUS	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
3D-LOTUS++	0$_{\pm0.0}$	**17**$_{\pm10.8}$	**30**$_{\pm13.4}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$
BridgeVLA (Ours)	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$	0$_{\pm0.0}$

Table 10 Per-task Success Rate on GemBench Level 4.

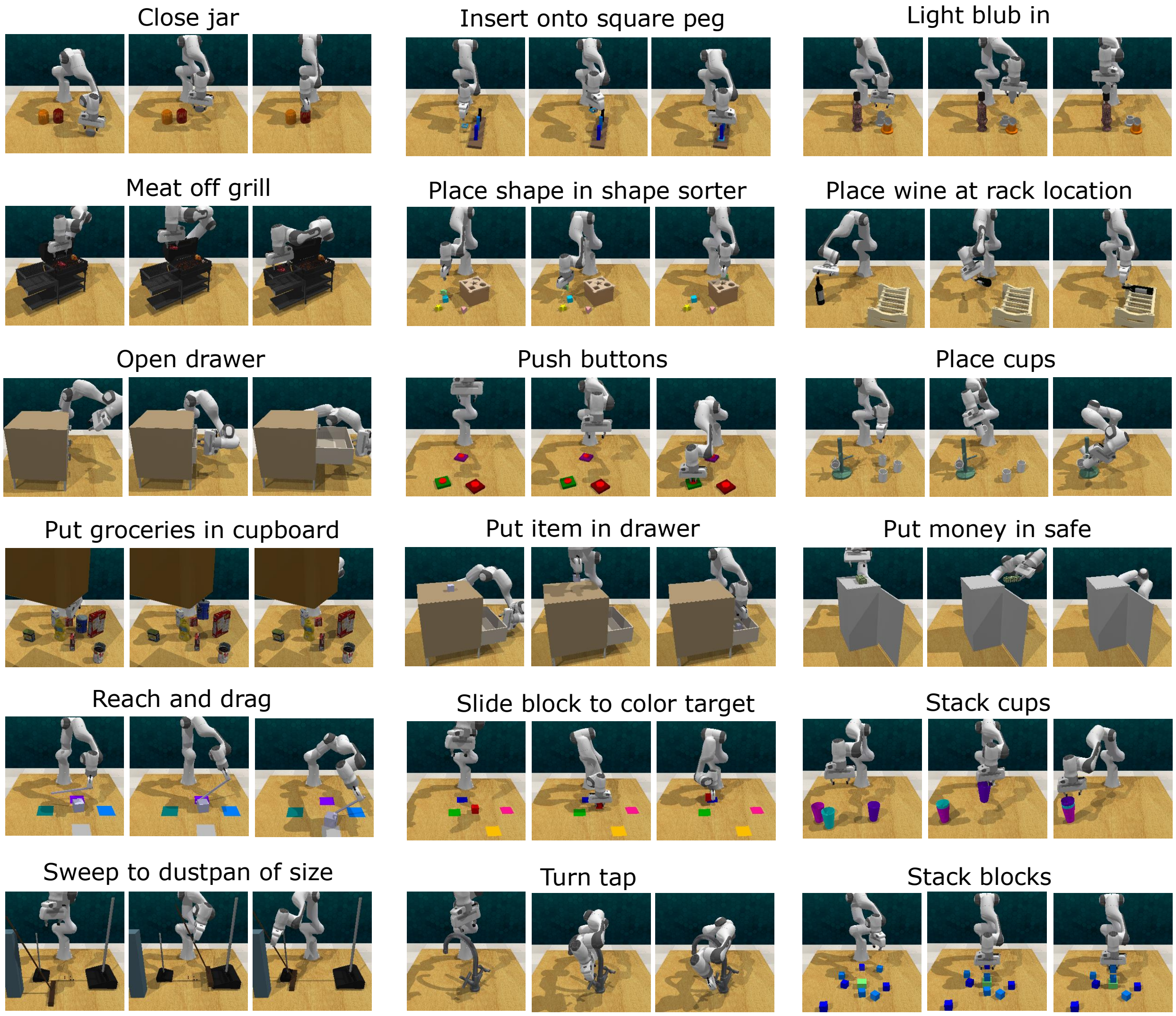


Figure 5 Visualization of 18 RLBench [17] Tasks.

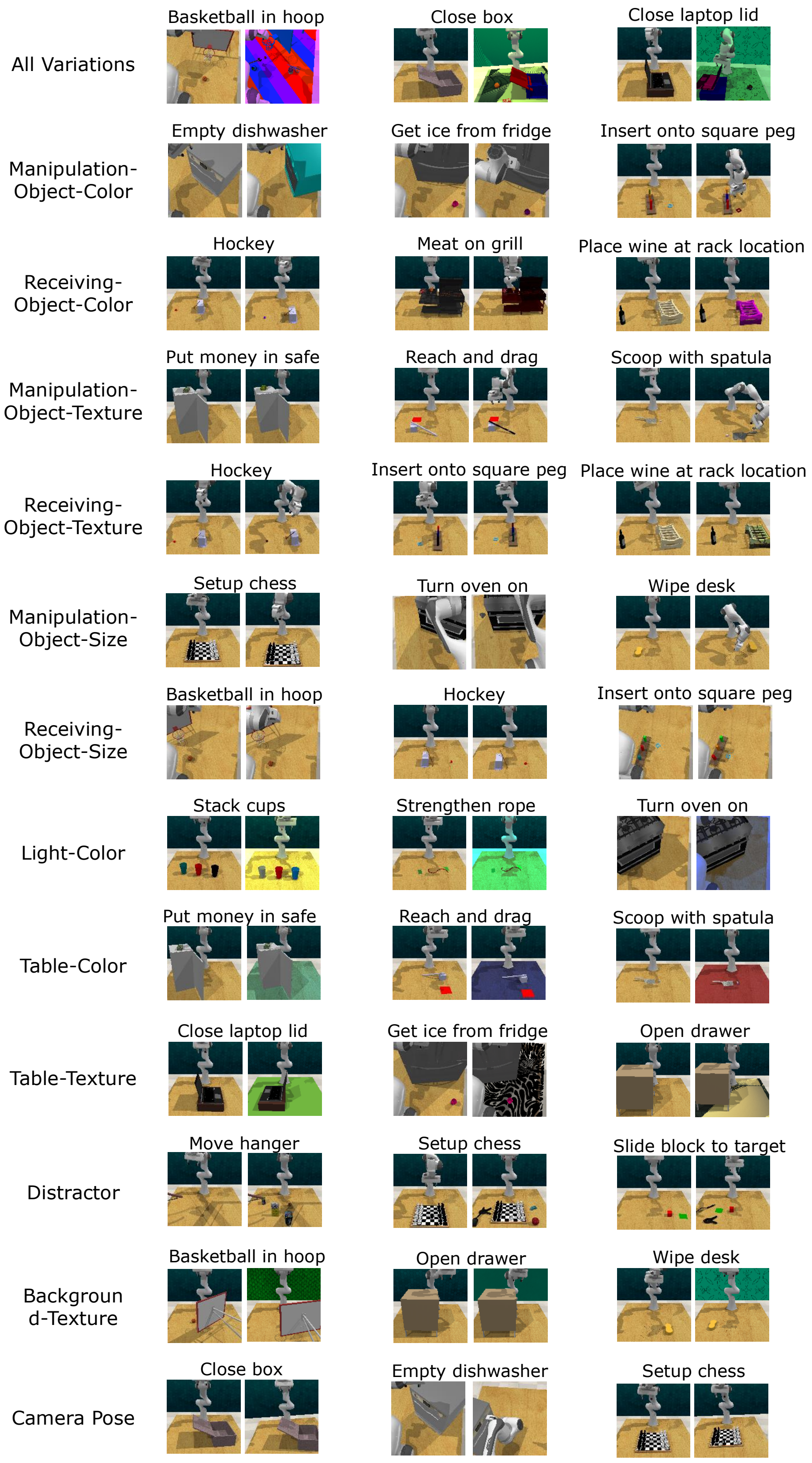


Figure 6 Visualization of Perturbations in COLOSSEUM [18].

Task	3 trajectories	10 trajectories
Put the RedBull can in the top shelf	5/5	5/5
Put the soda can in the bottom shelf	5/5	5/5
Put the RedBull can in the bottom shelf	5/5	5/5
Put the coke can in the top shelf	5/5	5/5
Place the red block in the blue plate	5/5	5/5
Place the orange block in the green plate	5/5	5/5
Put the wolf in the upper drawer	3/5	4/5
Place the red block in the purple plate	5/5	5/5
Place the yellow block in the green plate	5/5	5/5
Press sanitizer	5/5	5/5
Put the zebra in the upper drawer	5/5	5/5
Put the giraffe in the lower drawer	5/5	4/5
Put the zebra in the lower drawer	5/5	5/5

Table 11 Per-task Success Rates of BridgeVLA in the Basic Setting.

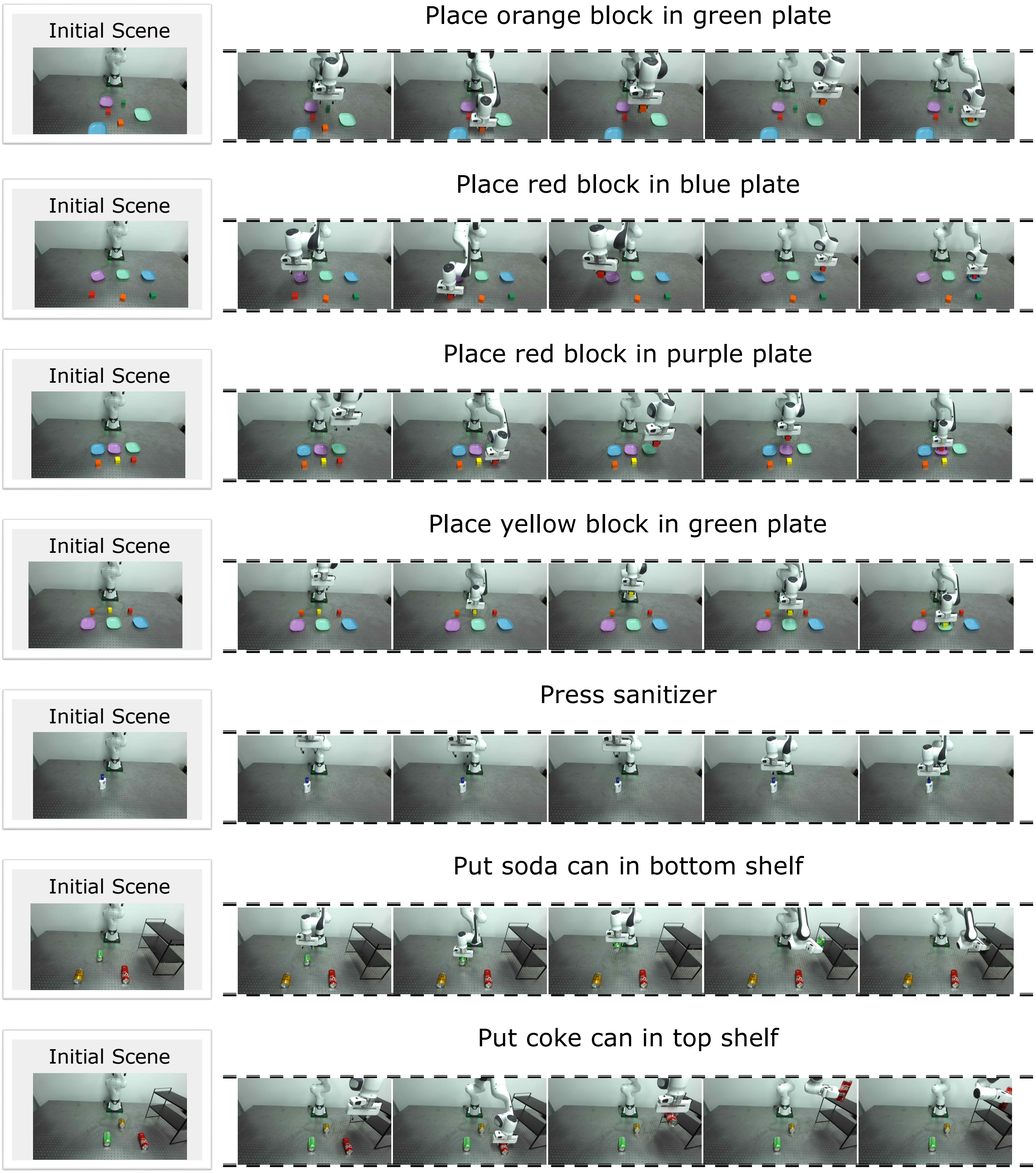

Initial Scene
Place orange block in green plate
Initial Scene
Place red block in blue plate
Initial Scene
Place red block in purple plate
Initial Scene
Place yellow block in green plate
Initial Scene
Press sanitizer
Initial Scene
Put soda can in bottom shelf
Initial Scene
Put coke can in top shelf

Figure 7 Real-Robot Rollouts (I).

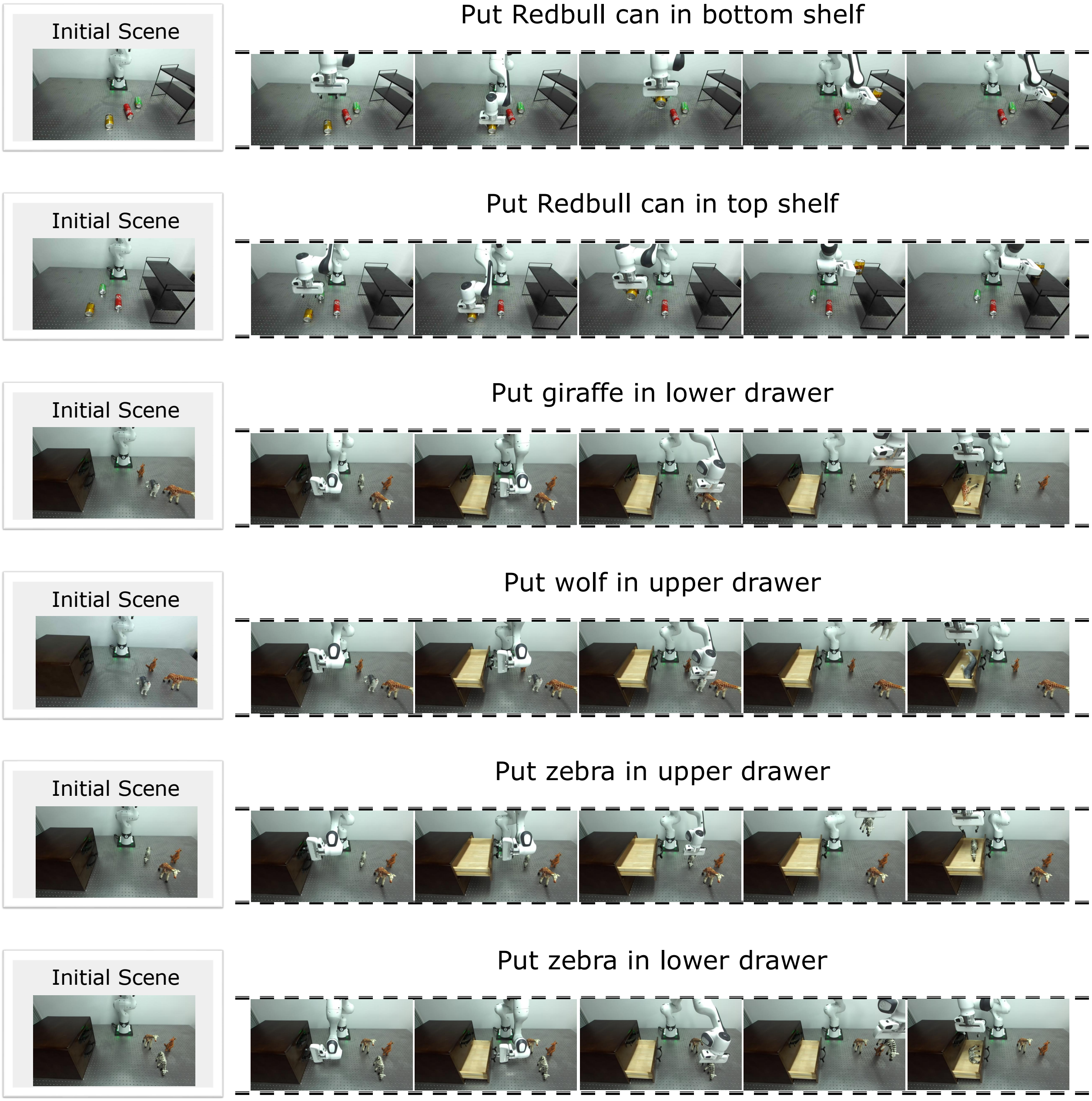
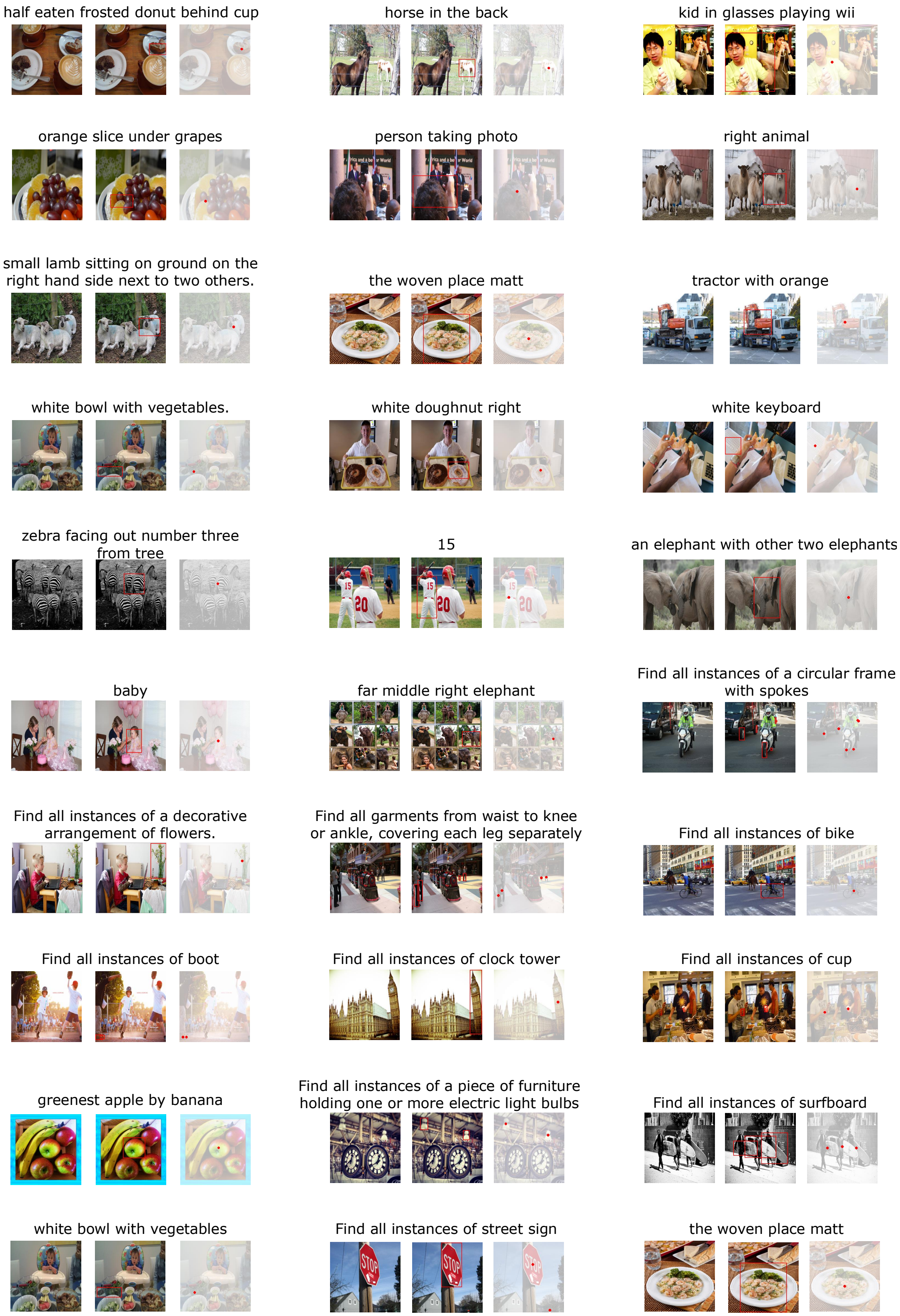

Initial Scene
Put Redbull can in bottom shelf
Initial Scene
Put Redbull can in top shelf
Initial Scene
Put giraffe in lower drawer
Initial Scene
Put wolf in upper drawer
Initial Scene
Put zebra in upper drawer
Initial Scene
Put zebra in lower drawer

Figure 8 Real-Robot Rollouts (II).



half eaten frosted donut behind cup

horse in the back

kid in glasses playing wii

orange slice under grapes

person taking photo

right animal

small lamb sitting on ground on the right hand side next to two others.

the woven place matt

tractor with orange

white bowl with vegetables.

white doughnut right

white keyboard

zebra facing out number three from tree

15

an elephant with other two elephants

baby

far middle right elephant

Find all instances of a circular frame with spokes

Find all instances of a decorative arrangement of flowers.

Find all garments from waist to knee or ankle, covering each leg separately

Find all instances of bike

Find all instances of boot

Find all instances of clock tower

Find all instances of cup

greenest apple by banana

Find all instances of a piece of furniture holding one or more electric light bulbs

Find all instances of surfboard

white bowl with vegetables

Find all instances of street sign

the woven place matt

Figure 9 Visualization of Pre-training Data. We list some samples of pre-training data. For every sample, the left shows the original image; the middle shows the bounding boxes of the objects of interest; the right shows the ground-truth heatmap used for training.

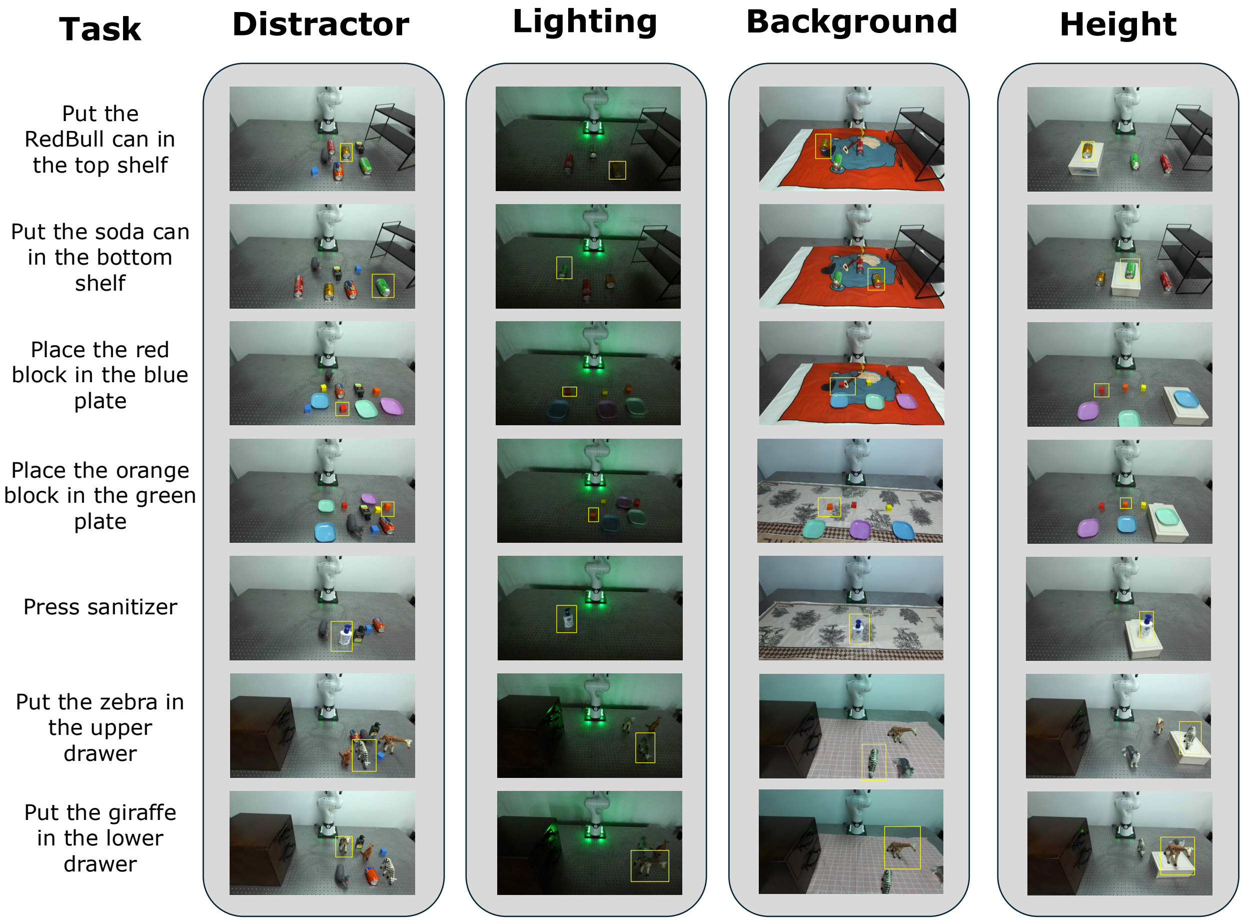

Task
Distractor
Lighting
Background
Height
Put the RedBull can in the top shelf
Put the soda can in the bottom shelf
Place the red block in the blue plate
Place the orange block in the green plate
Press sanitizer
Put the zebra in the upper drawer
Put the giraffe in the lower drawer

Figure 10 **Visualization of the Distractor, Lighting, Background, and Height settings.**

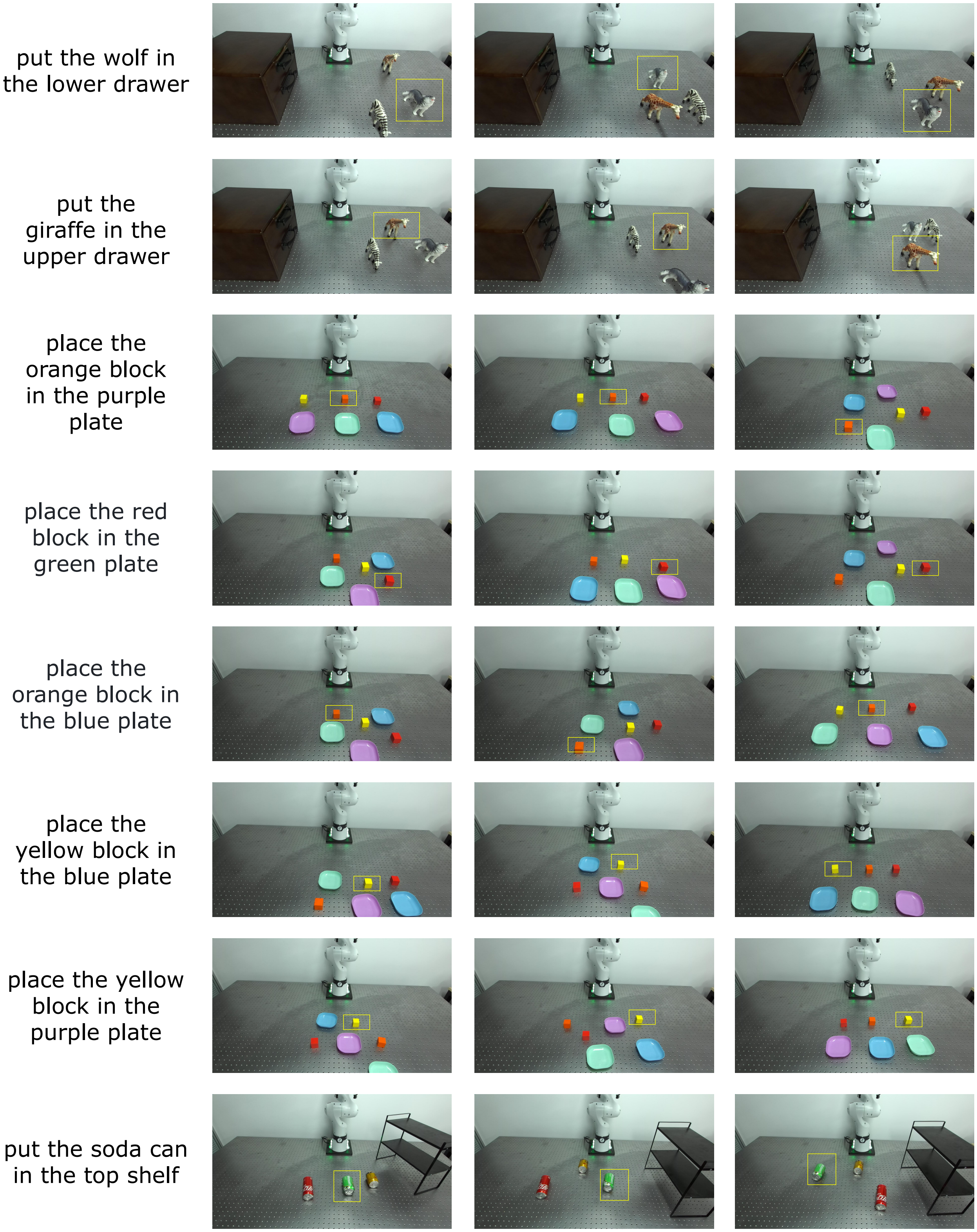

put the wolf in the lower drawer
put the giraffe in the upper drawer
place the orange block in the purple plate
place the red block in the green plate
place the orange block in the blue plate
place the yellow block in the blue plate
place the yellow block in the purple plate
put the soda can in the top shelf

Figure 11 Visualization of the Combination Setting (I). During training, the manipulated objects and skills are seen, but their combinations are unseen.

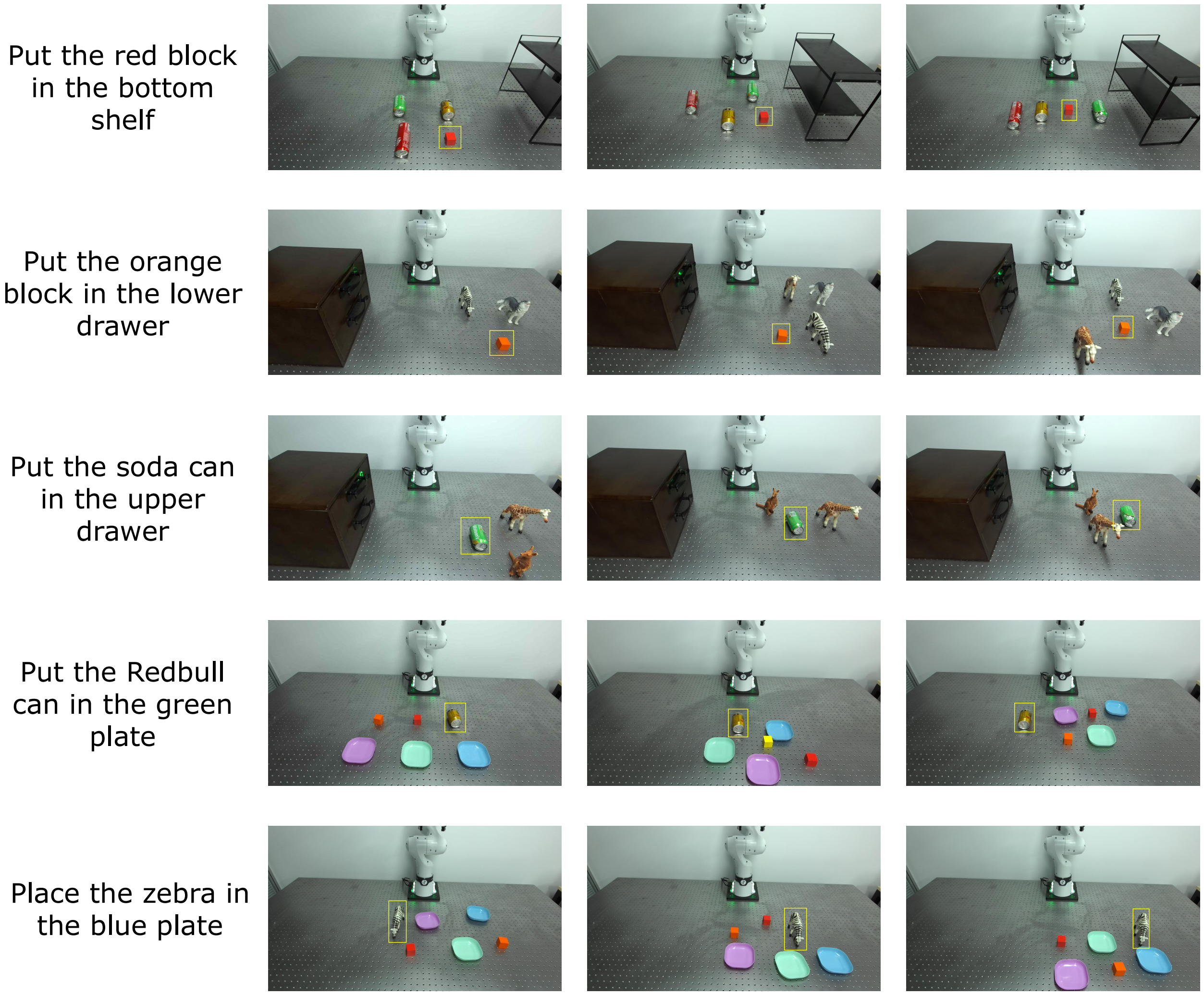

Put the red block in the bottom shelf
Put the orange block in the lower drawer
Put the soda can in the upper drawer
Put the Redbull can in the green plate
Place the zebra in the blue plate

Figure 12 Visualization of the Combination Setting (II). During training, the manipulated objects and skills are seen, but their combinations are unseen.

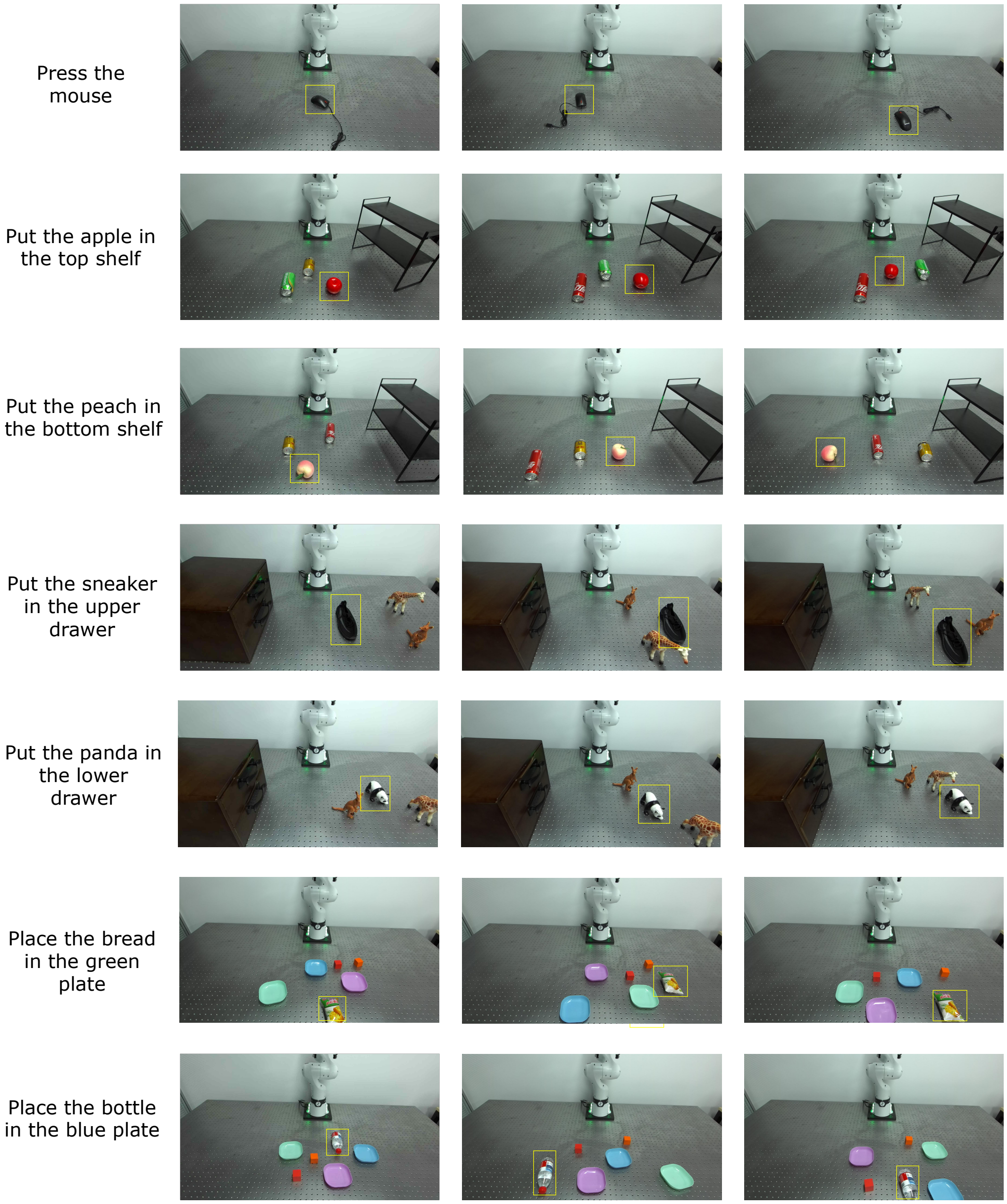

Press the mouse
Put the apple in the top shelf
Put the peach in the bottom shelf
Put the sneaker in the upper drawer
Put the panda in the lower drawer
Place the bread in the green plate
Place the bottle in the blue plate

Figure 13 Visualization of the Category Setting. In total, we evaluate on 7 objects from novel categories that are unseen during training.

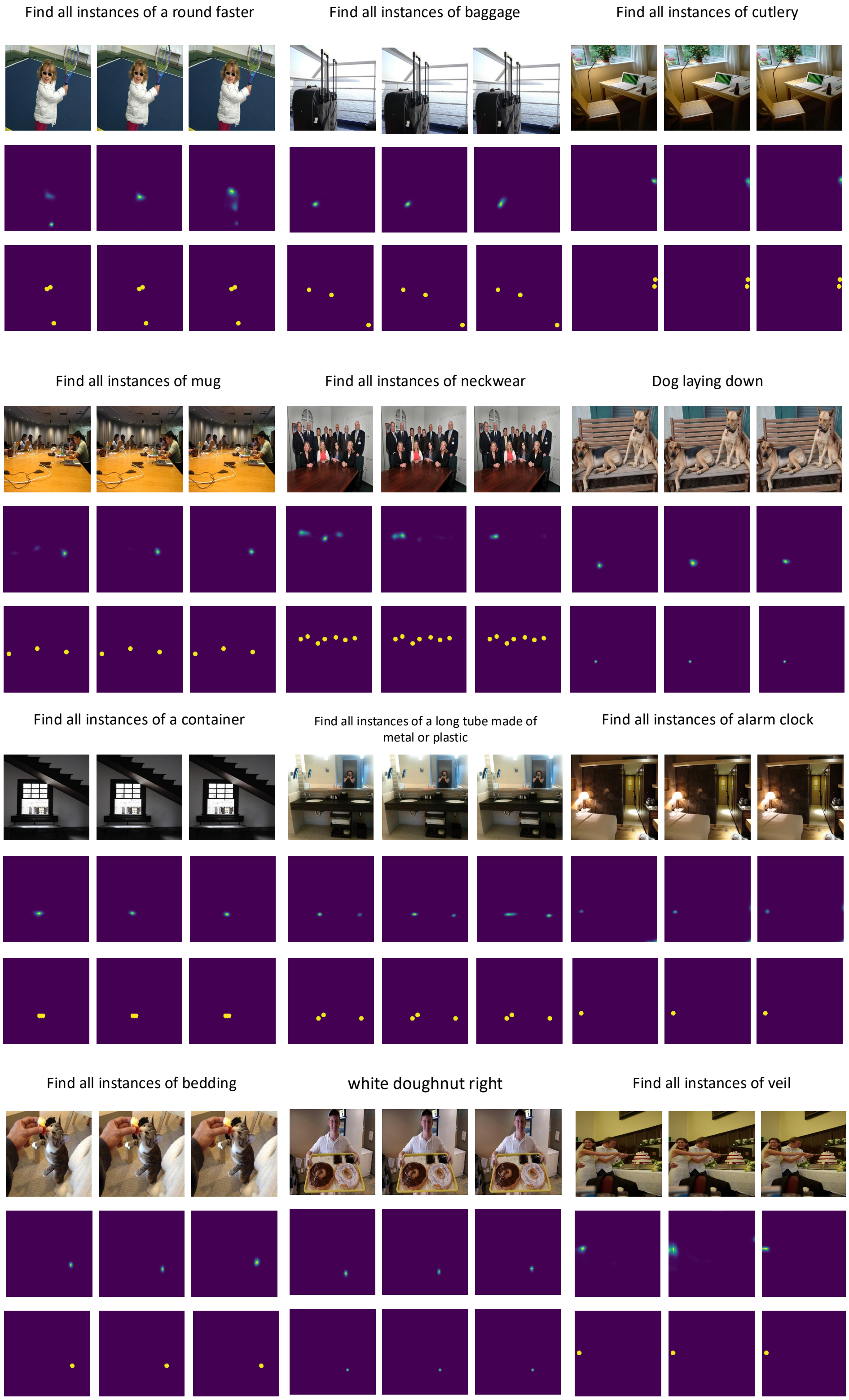

Find all instances of a round faster
Find all instances of baggage
Find all instances of cutlery
Find all instances of mug
Find all instances of neckwear
Dog laying down
Find all instances of a container
Find all instances of a long tube made of metal or plastic
Find all instances of alarm clock
Find all instances of bedding
white doughnut right
Find all instances of veil

Figure 14 Visualization of BridgeVLA's Prediction on Pre-training Dataset after Fine-tuning. To simulate the multi-view inputs during fine-tuning, we repeat the input image three times and feed them into the fine-tuned model to generate heatmaps. For each sample, the first row shows the input image; the second row shows the heatmap prediction; the third row shows the ground truth.