Robot Learning from Any Images

Siheng Zhao[1*], **Jiageng Mao**[1*], **Wei Chow**[1], **Zeyu Shangguan**[1], **Tianheng Shi**[1], **Rong Xue**[1], **Yuxi Zheng**[1], **Yijia Weng**[2], **Yang You**[2], **Daniel Seita**[1], **Leonidas Guibas**[2], **Sergey Zakharov**[3], **Vitor Guizilini**[3], **Yue Wang**[1]

[1]University of Southern California, [2]Stanford University, [3]Toyota Research Institute

Abstract: We introduce RoLA, a framework that transforms any in-the-wild image into an interactive, physics-enabled robotic environment. Unlike previous methods, RoLA operates directly on a single image without requiring additional hardware or digital assets. Our framework democratizes robotic data generation by producing massive visuomotor robotic demonstrations within minutes from a wide range of image sources, including camera captures, robotic datasets, and Internet images. At its core, our approach combines a novel method for single-view physical scene recovery with an efficient visual blending strategy for photorealistic data collection. We demonstrate RoLA's versatility across applications like scalable robotic data generation and augmentation, robot learning from Internet images, and single-image real-to-sim-to-real systems for manipulators and humanoids. Video results are available at our project page.

Keywords: Robotic Manipulation, Real-to-Sim-to-Real, Robotic Data Collection

"Each photograph is a portal to the physical world."

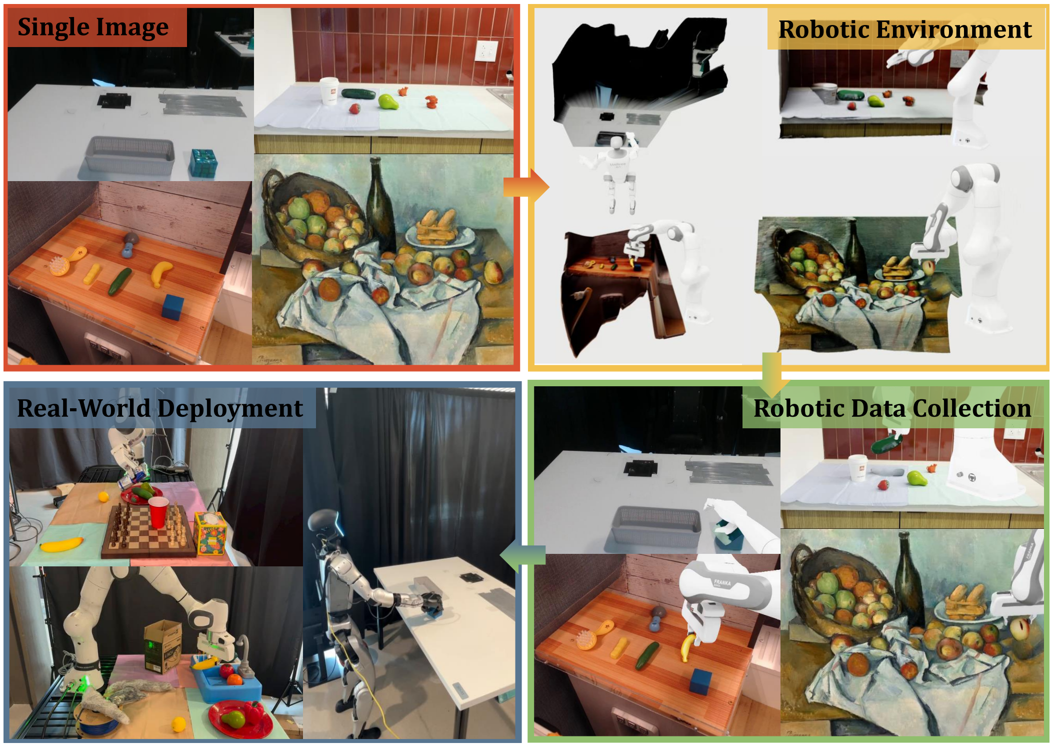


Figure 1: RoLA transforms a single in-the-wild image into an interactive, physics-enabled robotic environment. Given a single input image (**top-left**), RoLA recovers the physical scene for robot learning (**top-right**), enables large-scale robotic data generation (**bottom-right**), and supports deployment of learned policies on real robots (**bottom-left**).

[*] Equal contribution. Author order determined by coin flip.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Data quantity and diversity are widely recognized as primary bottlenecks in scaling robot learning. Although on-robot demonstrations provide physically accurate supervision, collecting them at scale demands specialized hardware and extensive human effort. Conversely, non-robotic visual data, such as images and videos, is virtually unlimited and encodes rich information relevant to robotic tasks. However, converting raw pixels into robot-complete data, *i.e.*, structured robotic scenes and demonstrations that can be directly used for policy learning, remains a fundamental challenge.

Real-to-sim-to-real methods [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] offer a promising direction toward bridging this gap. These approaches reconstruct robotic environments from multiview images or videos, simulate interactions to collect training data, and subsequently transfer the learned policies to real-world robots. While effective, their reliance on *ad hoc* hardware setups confines their data collection to laboratory settings, limiting their scalability to massive in-the-wild images that are readily available on the Internet or in the real world.

This paper poses a fundamental question: *Can we obtain robot-complete data from non-robotic images under minimal assumptions—ideally from a single image?* To answer this question, we revisit the underlying reasons why prior works rely on *ad hoc* camera setups to build robotic environments: (1) to reconstruct the scene and object geometry as simulation assets; (2) to generate photorealistic visual observations through physics-based or image-based rendering. Our key insight is that multiview reconstruction is not necessary for building robotic scenes, as with foundation model priors, a single RGB image is sufficient for recovering a physical scene. Furthermore, high-quality visual observations can be generated without full rendering pipelines, as visual blending of real images and virtual assets can also achieve high realism. Building on these insights, we eliminate the special need for hardware setups and make real-to-sim-to-real pipelines generalizable to arbitrary single images from anywhere without requiring additional information.

To this end, we introduce RoLA (Robot Learning from Any images), which automatically recovers a physical robotic scene configuration from a single image and enables scalable, photorealistic visuomotor data collection for policy learning. The framework consists of three key components: (1) Real-to-Sim: Recovers the physical scene from a single image by estimating object and scene geometry, inferring physical properties, establishing scene-object relationships, recovering camera pose, and placing the robot in appropriate locations; (2) Simulation: Generates large-scale simulated robotic trajectories in the recovered physical scene for diverse robotic tasks; (3) Sim-to-Real: Synthesizes photorealistic visuomotor demonstrations via visual blending for policy learning and deploys the learned policies to real robots. Together, these components form a fully automatic and generalizable pipeline that transforms any single image into a physically grounded, robot-interactable scene, bridging the gap between passive visual data and embodied robotic action.

By unleashing the power of in-the-wild images, RoLA unlocks a wide range of applications in real-world robotics. (1) Given a single image from a robotic camera, RoLA can recover the underlying physical scene and generate unlimited, physically plausible visuomotor demonstrations, providing abundant data for training imitation learning policies. (2) When paired with automatically generated language descriptions, RoLA further facilitates scalable data generation and augmentation for training vision-language-action (VLA) models. (3) RoLA also enables leveraging broader in-the-wild images, such as Internet photos, as priors for learning vision-based robotic manipulation. Using apple picking as a case study, we show that manipulation performance improves by incorporating priors learned from large-scale non-robotic visual data.

2 Related Works

Real-to-Sim-to-Real. Creating digital twins from real-world data offers a promising pathway to scale robot learning in simulation. Prior efforts have explored building digital robotic environments through multiview scene reconstruction [4, 8, 10, 3, 11, 12, 13, 6, 7, 14], object reconstruction [15, 16, 17, 18], procedural generation [19, 20, 21, 22, 23], or assets retrieval [1]. However, most works

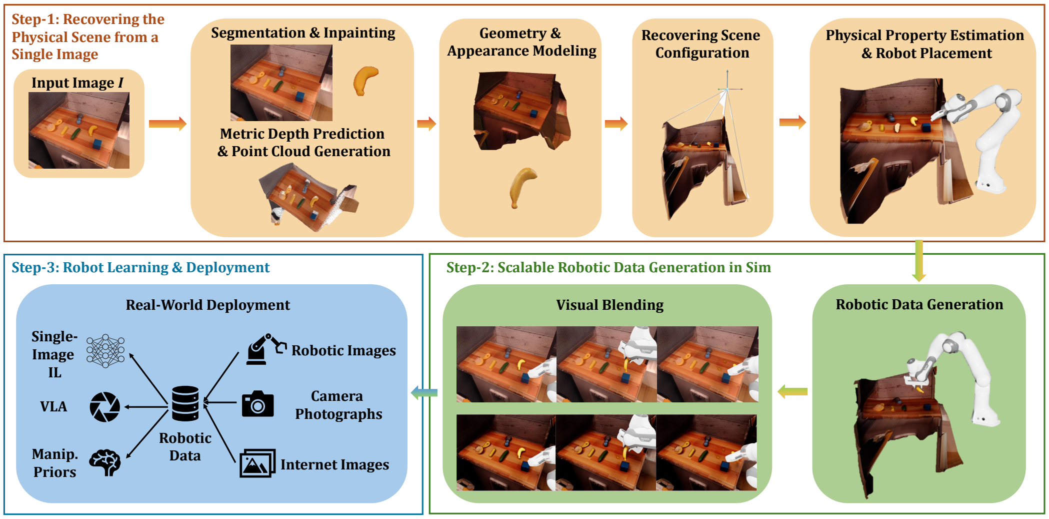

Step-1: Recovering the Physical Scene from a Single Image
Input Image I
Segmentation & Inpainting
Metric Depth Prediction & Point Cloud Generation
Geometry & Appearance Modeling
Recovering Scene Configuration
Physical Property Estimation & Robot Placement
Step-3: Robot Learning & Deployment
Real-World Deployment
Single-Image IL
VLA
Manip. Priors
Robotic Data
Robotic Images
Camera Photographs
Internet Images
Step-2: Scalable Robotic Data Generation in Sim
Visual Blending
Robotic Data Generation

Figure 2: An overview of the RoLA framework. **Step 1**: Recover the physical scene from a single image. **Step 2**: Generate large-scale photorealistic robotic demonstrations via visual blending. **Step 3**: Train and deploy policies across tasks and embodiments using the collected data.

need manual scene setup or require multiview camera captures, making them less generalizable to diverse real-world data. Notably, some works [1, 2] focus on recovering the 3D scene from a single image. Nonetheless, these works typically retrieve existing assets, thus constraining generalization to the scope of pre-existing databases. In contrast, our approach is fully automatic and operates on any single image without requiring additional information or external databases.

Robot Learning from Unstructured Data. Unstructured Internet images and videos offer a vast but underutilized resource for training robot policies. Prior work extracts actionable knowledge from videos by learning priors from human-object interactions [24, 25, 26], motion retargeting [27, 28], inverse dynamics [29, 30], and affordance transfer [31, 32]. Notably, recent works resort to pixel-level augmentations, e.g., changing backgrounds [33], viewpoints [34], embodiments [35, 36, 37], and green screens [38, 5, 39], to synthesize diverse visual demonstrations. However, these pixel editing approaches cannot ensure physical accuracy in generating robotic demonstrations. In contrast, our approach combines physics simulation and visual blending, which generates both physically accurate and photorealistic robotic demonstrations.

Physics-based Scene Generation. Several recent works [40, 41, 42, 43] explore generating a physically interactable digital twin from a single image, typically for physics-based video generation [44, 45]. However, these works generally focus on narrow object categories and show limited applications in robotics. In contrast, our approach is generalizable to diverse images and objects, and fully supports robotic simulation and data generation.

3 Method

We study how to leverage a single image from any source as a robotic environment for scalable robotic data generation and policy learning. We achieve this goal by first recovering the physical scene behind the input image (Section 3.1), then collecting visuomotor robotic demonstrations in the physical scene (Section 3.2), and finally training robotic policies and deploying them to the real robots (Section 3.3). An overview of our method is shown in the Figure 2.

3.1 Recovering the Physical Scene from a Single Image

Problem Definition. We consider the inverse problem of how a real image is created. The forward image formation process can be expressed as:

$$I = \pi(S; \mathcal{C}), \tag{1}$$

where $I \in \mathbb{R}^{H \times W \times 3}$ is the observed image, and π is the camera projection function. The physical scene $S = \{O, B\}$ consists of objects O and a background B, each represented as a textured mesh $\mathcal{M}$ with central position $\mathbf{p}$, orientation $\mathbf{q}$ (e.g., quaternion), and physical properties $\mathcal{P}$ (such as mass and friction). The camera parameters $\mathcal{C}$ include intrinsics $\mathbf{K}$ and extrinsics $[\mathbf{R} \mid \mathbf{t}]$.

Our goal is to recover the entire physical scene $\{\mathcal{M}, \mathcal{P}, \mathbf{p}, \mathbf{q}, \mathcal{C}\}$ from a single image I. This inverse problem is inherently ill-posed. However, we demonstrate that with appropriate modeling and priors, it is possible to obtain accurate and robust estimates of the underlying scene structure.

Geometry and Appearance Modeling. We begin by obtaining textured meshes $\mathcal{M}$ for both the objects O and the background B, which recovers the scene geometry and appearance.

For object modeling, we first generate object segmentation masks from I using the grounded segmentation model [46]. The segmented object regions are then passed to a state-of-the-art image-to-3D generation model [47], which produces the object textured mesh $\mathcal{M}_O$. For articulated objects, since single-view articulated reconstruction remains an open problem, we retrieve similar assets from existing 3D model repositories [48, 49] as a practical alternative.

For background modeling, we use an image inpainting model [50] to synthesize a background-only image I_B, which is used to texture the background mesh. To estimate background geometry, we apply a metric depth prediction model [51] to infer an absolute depth map D and camera intrinsics $\mathbf{K}$ from I. Using inverse projection, we construct a scene point cloud $P_S \in \mathbb{R}^{HW \times 3}$:

$$\mathbf{X}_{uv} = D(u, v) \cdot \mathbf{K}^{-1}[u, v, 1]^{\top}, \tag{2}$$

where (u, v) are image coordinates and $\mathbf{X}_{uv}$ is the corresponding 3D point in the camera frame.

We then partition P_S into object and background point clouds, P_O and P_B, using the object segmentation masks. A background mesh $\mathcal{M}_B$ is reconstructed from P_B via surface reconstruction techniques [52]. Since the background primarily serves as a support surface in robotic tasks, we optionally approximate it using shape primitives (e.g., planes) to accelerate physics simulation.

We note that the vision foundation models [51, 46, 47, 50] exhibit strong generalization to diverse image types, enabling our method to handle in-the-wild visual data robustly.

Recovering Scene Configuration. Given the textured meshes $\mathcal{M} = \{\mathcal{M}_O, \mathcal{M}_B\}$ for the object and background, our goal is to recover the full scene configuration: the sizes, positions $\mathbf{p}$, and orientations $\mathbf{q}$ of the meshes, as well as the camera pose $[\mathbf{R} \mid \mathbf{t}]$ in a gravity-aligned world frame.

We leverage the scene point cloud P_S, derived from metric depth, as the reference geometry, since it inherently encodes both the scene's spatial structure and its absolute scale. The background mesh $\mathcal{M}_B$, reconstructed directly from the background point cloud $P_B \subset P_S$, preserves its real-world scale and alignment. For the object mesh $\mathcal{M}_O$, which is generated independently from image-to-3D models, we rescale it and register its position $\mathbf{p}$ and orientation $\mathbf{q}$ to the corresponding object point cloud P_O using mesh-to-point registration [53], which establishes the scene-object relationships.

To place the scene and camera in a physically meaningful world frame, we introduce a concept named the *supported plane*, which is either the tabletop or the ground plane. We assume the *supported plane* is always perpendicular to the Z-axis, i.e., the gravity axis in physics simulation. With this assumption, we first run a ground segmentation [46] and a RANSAC plane estimation algorithm on the scene point cloud P_S to estimate the ground normal direction $\mathbf{n}$. We then estimate the rotations $\mathbf{R}$ of the scene and camera that maps the plane normal to the Z-axis $\mathbf{z} = [0, 0, 1]^{\top}$:

$$\mathbf{R} \ = \ \mathbf{I}_3 + \sin\theta \, [\mathbf{k}]_{\times} + (1 - \cos\theta) \, [\mathbf{k}]_{\times}^2, \tag{3}$$

where $[\mathbf{k}]_{\times}$ is the 3×3 skew-symmetric matrix of $\mathbf{k} = \frac{\mathbf{n} \times \mathbf{z}}{\|\mathbf{n} \times \mathbf{z}\|_2}$, and $\theta = \arccos(\mathbf{n}^{\top}\mathbf{z})$ following the Rodrigues' formula. Finally, we rotate the scene and update the camera position $\mathbf{t}$ to center the scene at the origin of the world frame. This results in the full camera pose $[\mathbf{R} \mid \mathbf{t}]$ in the world frame.

This step fully recovers the spatial configuration of object, background, and camera, enabling physically plausible visuomotor demonstration generation from the original image layout and viewpoint.

Physical Property Estimation. To enable physical interaction, we estimate key physical properties $\mathcal{P}$, such as mass (or density) and friction coefficients, required by physics simulators. To get initial estimates for different objects in a scalable way, we leverage large language models (LLMs) [54] to provide physical property estimates based on the object category and image context. Specifically, we first identify the object class, then prompt the LLM with the class name and visual context to infer plausible physical parameters. We empirically found that the estimation is quite robust to support the subsequent real-world policy deployment (see Section 4.3).

Robot Placement. To convert the physical scene into a robotic environment, we must determine appropriate robot positions. For images captured by a robot-mounted camera, we directly use the known camera-to-robot transformation to place the robot in the correct pose. For non-robotic images, we propose a sampling-based method to generate feasible placements. We approximate the robot's reachable workspace as a 3D shell and sample base positions such that this region covers the axis-aligned bounding boxes of all objects. To avoid collisions, we discard placements that intersect with the scene's bounding box. This process produces multiple valid placement candidates, enabling diverse action trajectory generation from different robot positions.

3.2 Robotic Data Generation

Data Collection. We support multiple modes of collecting robotic demonstrations within our image-generated environment. These include direct control of robots in physics simulators via keyboard or space mouse, scripted policies using motion planners [55], and pretrained manipulation policies [56, 57]. These methods enable efficient large-scale robotic data collection; for example, over 1,500 visuomotor demonstrations can be collected within an hour using an RTX4090 GPU. For each demonstration, we record T frames of robot actions $\{a_t\}_{t=0}^T$, depth maps $\{D_t\}_{t=0}^T$, and visual observations $\{I_t\}_{t=0}^T$ from the original camera viewpoint, which are later used to compose photorealistic visuomotor demonstrations.

Visual Blending. A key challenge in sim-to-real adaptation is the visual domain gap: rendered observations from simulators often differ significantly in appearance from real-world images. To mitigate this, we introduce a visual blending technique that enhances the photorealism of rendered visual observations by combining them with the real background image I_B using z-buffering.

Let I_B and D_B be the background image and depth, and $\{I_t\}_{t=0}^T$ and $\{D_t\}_{t=0}^T$ denote the rendered images and depth maps in simulation. The blended observations $\{I'_t\}_{t=0}^T$ can be computed as

$$I'_t = M_t \odot I_t + (1 - M_t) \odot I_B, \quad \text{where} \quad M_t(u,v) = \begin{cases} 1, & \text{if } D_t(u,v) < D_B(u,v) - \epsilon, \\ 0, & \text{if } D_t(u,v) \geq D_B(u,v) - \epsilon. \end{cases} \tag{4}$$

The binary mask $M_t \in \{0,1\}^{H \times W}$ indicates where the rendered surface is closer to the camera than the real background, ϵ is a small threshold for depth noise and alignment error, and $\odot$ denotes element-wise multiplication. This blending preserves the real image in background regions while inserting rendered content such as objects and robots only where it appears physically in front of the scene. Compared to [38], visual blending with z-buffering handles the occlusion cases more effectively, resulting in more photorealistic and consistent visual observations. We store the resulting blended observations along with the corresponding robot actions $\{I'_t, a_t\}_{t=0}^T$ as visuomotor demonstrations for downstream policy training.

3.3 Robot Learning & Deployment

The image-generated robotic environments enable scalable data generation across a wide range of tasks and robotic embodiments. Using the collected visuomotor demonstrations $\{I'_t, a_t\}_{t=0}^T$, we train visuomotor policies in simulation and seamlessly deploy them to the real world. Our framework can also be used as a scalable data generator to augment existing robotic datasets [58] by conditioning on their images as inputs. Furthermore, by automatically generating language annotations for visual demonstrations using LLMs, our method supports scalable robotic data generation for training more generalizable vision-language-action models (VLAs).

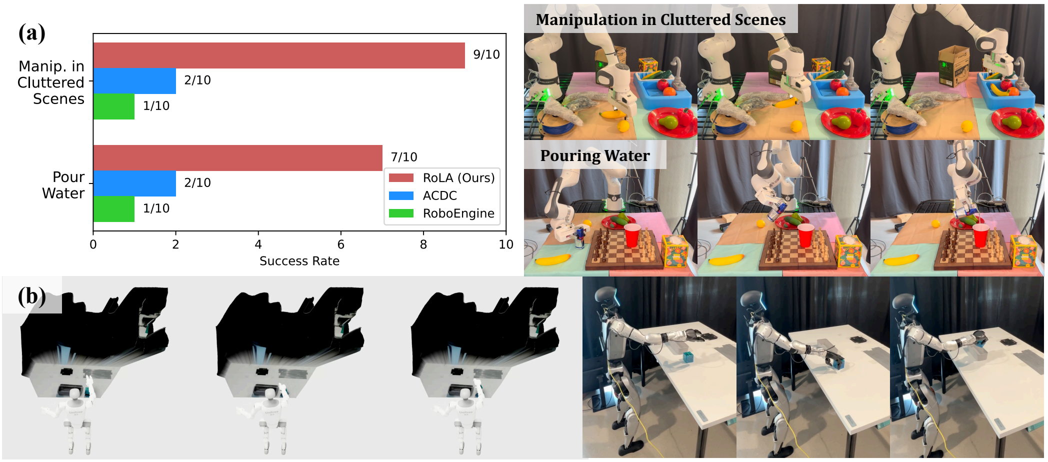


Figure 3: (a) Real-world deployment of policies trained with RoLA-generated data. (b) RoLA enables efficient real-to-sim-to-real transfer for humanoid robots.

4 Experimental Evaluation

In this section, we investigate the following questions through extensive experiments:

Q1: Can RoLA recover an accurate physical scene from a single image that enables effective robotic policy learning, achieving performance comparable to multiview reconstruction methods?
Q2: Can RoLA efficiently generate large-scale, high-quality robotic data? How does the quality of the generated data compare to previous retrieval-based and pixel-editing-based methods?
Q3: Can RoLA enable training of visuomotor policies and vision-language-action (VLA) models from single images, generalize across different embodiments (*e.g.*, manipulators and humanoid robots), and deploy successfully to the real world?
Q4: Can RoLA leverage in-the-wild Internet images for vision-based robot learning?
Q5: How does visual blending improve RoLA's performance?

Due to the page limit, we refer readers to the appendix for detailed experimental setups.

4.1 Physical Scene Recovery

A common concern of single-image scene recovery is the potential inaccuracy of scene properties due to the lack of geometric constraints. We investigate this issue by comparing our method with a multiview reconstruction baseline [4]. Specifically, we build digital twins of robotic environments using both the RoLA pipeline and [4] on the task *"pick up the banana and put it onto the stove"*, collect 200 demonstrations respectively, train an imitation learning policy [59], and deploy it for robotic manipulation. As shown in Table 1, RoLA's success rate closely matches the multiview reconstruction baseline, indicating that a single image is sufficient for creating a realistic robotic environment for policy learning (**Q1**). Although multiview reconstruction methods achieve slightly better performance ($\approx 3.3\%$), their reliance on controlled setups makes them significantly less scalable than our approach, which operates on single images without additional requirements.

Method	**Multi-view**	**Single-view (RoLA)**
Success Rate	$75.5 \pm 7.5\%$	$72.2 \pm 2.1\%$

Table 1: Comparison of policy success rates between multi-view reconstruction and our single-view RoLA pipeline. Average success rates ± StdErr are calculated over 3 seeds.

4.2 Robotic Data Generation

To evaluate RoLA's data generation quality, we set up a robotic manipulation benchmark and compare policies trained on RoLA-generated data with those from baseline methods: (1) ACDC [1], a retrieval-based single-image data generation method, and (2) RoboEngine [33], an image-based demonstration augmentation approach. For each task, we collect 200 visuomotor demonstrations

Language Instruction	Success Rate	Language Instruction	Success Rate
reach the blue cup	10/10	pick up the carrot	10/10
put the green pepper beside the lemon	9.5/10	put the yellow lemon beside the green apple	10/10
take the banana and place it beside the potato	8/10	pick up the red tomato and place it beside the yellow lemon	9/10
take the grey object beside the lemon and place it beside the yellow apple	6.5/10	take the strawberry off the potato and place it beside the red apple	10/10

Table 3: Simulation evaluation of our VLA model trained on RoLA-generated data. Each task is tested over 10 trials with randomized object poses, under SimplerEnv [38] setting. Note that *partial success* (score of 0.5) is possible for tasks following OpenVLA [60]. See Appendix for more details.

using RoLA and each baseline, train imitation learning policies [59], and measure success rates. As shown in Table 2, RoLA significantly outperforms the baselines across tasks, indicating that data generation from recovered physical scenes yields more faithful and physically accurate demonstrations than retrieval or pixel-editing methods (**Q2**).

4.3 Single-Image Robot Learning

We conduct real robot experiments to investigate whether RoLA can enable single-image real-world robot learning. Specifically, we employ a Franka Research 3 Robot as the tested embodiment.

Task	RoLA (Ours)	RoboEngine	ACDC
Broccoli into Bowl	$53.4 \pm 11.7\%$	$8.3 \pm 11.2\%$	$25.1 \pm 17.5\%$
Banana onto Stove	$85.4 \pm 11.6\%$	$3.3 \pm 4.7\%$	$0.0 \pm 0.0\%$
Carrot onto Burner	$90.0 \pm 10.8\%$	$22.7 \pm 16.6\%$	$6.4 \pm 9.0\%$
Average	$\mathbf{76.2 \pm 19.8\%}$	$11.4 \pm 13.4\%$	$10.5 \pm 14.9\%$

Table 2: Comparison of imitation learning success rates using data generated by RoLA versus baselines. Average success rates ± StdErr are calculated over 3 seeds.

We set up 2 real-world robotic manipulation tasks across diverse environments: (1) Object manipulation in cluttered scenes: the robot must pick up a banana in a cluttered environment and place it accurately inside a sink, evaluating grasp accuracy and collision avoidance. (2) Pouring water: The robot must pour soda from a can into a cup, evaluating manipulation precision. For each task, we capture an RGB image of the scene using a RealSense D415 camera as input to RoLA, then collect 200 demonstrations using our framework, train a diffusion policy, and deploy it to the real robot. During evaluation, we test each task for 10 trials. In each trial, the manipulated object is randomly placed within a 6cm×6cm workspace with random poses. As shown in Figure 3(a), the policies trained with RoLA can be successfully deployed to the real world with high success rates.

To further investigate whether RoLA supports different embodiments, *e.g.*, a humanoid robot, we leverage a Unitree G-1 robot as the tested embodiment. We leverage a photo captured by the humanoid's head-mounted camera as input to RoLA. Through the RoLA pipeline, we reconstruct the physical scene from the ego-centric image, control the robot to grasp a blue cube with a dexterous hand in the recovered scene, and deploy the actions to the real robot to complete the same task. As shown in Figure 3(b), the humanoid robot successfully finished the task, demonstrating RoLA's support for different robot types. Experiments on both robotic manipulators and humanoid robots have demonstrated that RoLA effectively supports single-image, real-world robot learning (**Q3**).

4.4 Training Vision-Language-Action Models with RoLA-Generated Data

RoLA supports highly efficient and scalable robotic data generation from single images. We investigate whether this can facilitate large-scale training of vision-language-action (VLA) models. To this end, we collect over 60,000 demonstrations (the same scale of BridgeData V2 [58]), and train a VLA model from scratch using Qwen2.5 [61] VLM as the backbone, following the MiniVLA [62] training setting. Training occurs on 8×H100 GPUs over four days (i.e. 768 GPU-hours).

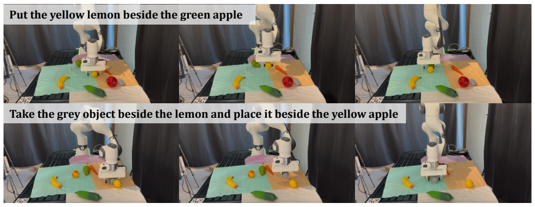

Put the yellow lemon beside the green apple
Take the grey object beside the lemon and place it beside the yellow apple

Figure 4: Real-world execution of a VLA model trained with RoLA-generated data.

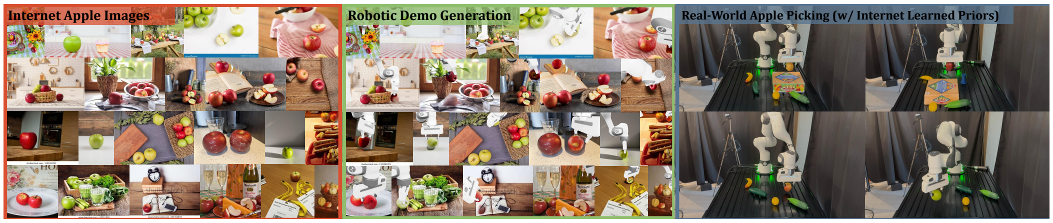


Figure 5: Learning a vision-based apple grasping prior from Internet apple images.

To evaluate our VLA model's performance, we design 10 evaluation tasks under the widely-used SimplerEnv [38] setting, testing skill diversity, scene variations, object category and pose variations, semantic understanding, and complex visual reasoning. Each task is evaluated over 10 trials with randomly initialized object poses. Table 3 illustrates our model's strong generalization from purely RoLA generated data. We then successfully deploy the VLA model to real-world scenarios, with example rollouts showcased in Figure 4.

4.5 Learning a Vision-Based Grasping Prior from Internet Images

We explore the potential of more in-the-wild data, such as Internet images, as a pretraining source for robot learning. Due to resource limitations, we only chose the apple grasping task as a case study. Specifically, we collect 2,000 Internet apple images and leverage RoLA to generate over 3,000 demonstrations to pick up apples in the Internet images (See Figure 5). We pre-train a neural network that encodes images and outputs the 6-DoF grasping pose in the image 3D frame. Since Internet images contain apples of various sizes, types, positions, and with diverse backgrounds and viewpoints, we hope the pretrained network can learn a generalizable vision-based apple grasping prior from the Internet images. During deployment, we finetune the network with 10/20/50 real demonstrations and test the real-world apple grasping success rate with different apple types, positions, and camera viewpoints. As shown in Table 4, the grasping success rate significantly increases with the learned grasping prior, indicating the potential of pretraining with Internet images for robot learning (**Q4**).

Method \# of Demos	**10**	**20**	**50**
w/ grasping prior	2/10	3/10	8/10
w/o grasping prior	0/10	3/10	3/10

Table 4: Real-world apple grasping success across varying numbers of fine-tuning demonstrations, w/ and w/o the Internet-trained grasping prior.

4.6 Empirical Study

We investigate how the proposed visual blending affects the learning performance. In particular, we follow the training setting in Section 4.2 but replace visual blending with rendering to generate visuomotor demonstrations. As shown in Table 5, visual blending significantly improves the visual quality of demonstrations and enhances the learning performance (**Q5**).

Task	**Robot**	**w/ Blending**	**w/o Blending**
Pickplace Banana	WidowX	80.0 ± 17.8%	33.3 ± 23.5%
Spatula onto Board	WidowX	85.0 ± 4.0%	6.7 ± 9.4%
Broccoli into Bowl	WidowX	53.4 ± 11.7%	0.0 ± 0.0%
Pickplace Spoon	Franka	100.0 ± 0.0%	22.2 ± 22.0%
Banana onto Stove	Franka	85.4 ± 11.6%	11.7 ± 16.5%
Carrot onto Burner	Franka	90.0 ± 10.8%	0.0 ± 0.0%

Table 5: Ablation study comparing visual blending with direct mesh rendering (w/o blending). Visual blending yields better performance across both WidowX and Franka robots.

5 Conclusion

We presented RoLA, a framework that transforms a single image into an interactive, physics-enabled robotic environment for scalable data generation and policy learning. RoLA faithfully recovers physical scenes from diverse image sources, enabling the collection of photorealistic, physically plausible demonstrations without multiview captures. Through extensive robotic experiments, we demonstrate that RoLA supports single-image robot learning, real-world deployment, and training of vision-language-action models. Our results highlight the potential of leveraging vast in-the-wild visual data to advance scalable, generalizable robot learning.

6 Limitations

This work has several limitations. (1) The quality of the generated robotic demonstrations is bounded by the fidelity of the physics simulator. However, we emphasize that the core technical contribution of this paper lies in a highly scalable and generalizable approach for transforming massive non-robotic visual data into physically plausible robotic data. While the transformed demonstrations may not match the precision of real-world collected data, our framework can generate data several orders of magnitude faster and at a much larger scale than real-world collection. By leveraging the virtually unlimited pool of Internet images, RoLA supports data generation with high diversity in object types, scene layouts, appearances, and viewpoints, significantly expanding the scalability and generalization potential of robot learning. (2) While our method does not support changing camera viewpoints within a single scene as multiview reconstruction methods do, it compensates by leveraging Internet images captured from diverse natural viewpoints. As a result, RoLA can generate training data with rich viewpoint diversity across scenes, enabling the learned policies to generalize well to novel camera perspectives.

7 Acknowledgements

The USC Geometry, Vision, and Learning Lab acknowledges generous supports from Toyota Research Institute, Dolby, Google DeepMind, Capital One, Nvidia, and Qualcomm. Yue Wang is also supported by a Powell Research Award.

References

[1] T. Dai, J. Wong, Y. Jiang, C. Wang, C. Gokmen, R. Zhang, J. Wu, and L. Fei-Fei. Automated creation of digital cousins for robust policy learning. *CoRL*, 2024.

[2] A. Zook, F.-Y. Sun, J. Spjut, V. Blukis, S. Birchfield, and J. Tremblay. Grs: Generating robotic simulation tasks from real-world images. *arXiv preprint arXiv:2410.15536*, 2024.

[3] X. Han, M. Liu, Y. Chen, J. Yu, X. Lyu, Y. Tian, B. Wang, W. Zhang, and J. Pang. Re3 sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation. *arXiv preprint arXiv:2502.08645*, 2025.

[4] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *RSS*, 2024.

[5] Y. Fang, Y. Yang, X. Zhu, K. Zheng, G. Bertasius, D. Szafir, and M. Ding. Rebot: Scaling robot learning with real-to-sim-to-real robotic video synthesis. *arXiv preprint arXiv:2503.14526*, 2025.

[6] H. Lou, Y. Liu, Y. Pan, Y. Geng, J. Chen, W. Ma, C. Li, L. Wang, H. Feng, L. Shi, et al. Robogs: A physics consistent spatial-temporal model for robotic arm with hybrid representation. *ICRA*, 2025.

[7] X. Li, J. Li, Z. Zhang, R. Zhang, F. Jia, T. Wang, H. Fan, K.-K. Tseng, and R. Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator. *arXiv preprint arXiv:2411.11839*, 2024.

[8] N. Pfaff, E. Fu, J. Binagia, P. Isola, and R. Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups. *arXiv preprint arXiv:2503.00370*, 2025.

[9] W. Ye, F. Liu, Z. Ding, Y. Gao, O. Rybkin, and P. Abbeel. Video2policy: Scaling up manipulation tasks in simulation through internet videos. *arXiv preprint arXiv:2502.09886*, 2025.

[10] S. Patel, X. Yin, W. Huang, S. Garg, H. Nayyeri, L. Fei-Fei, S. Lazebnik, and Y. Li. A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards. *ICRA*, 2025.

[11] S. Zhu, L. Mou, D. Li, B. Ye, R. Huang, and H. Zhao. Vr-robo: A real-to-sim-to-real framework for visual robot navigation and locomotion. *arXiv preprint arXiv:2502.01536*, 2025.

[12] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, 2017.

[13] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018.

[14] W. Peng, J. Lv, Y. Zeng, H. Chen, S. Zhao, J. Sun, C. Lu, and L. Shao. Tiebot: Learning to knot a tie from visual demonstration through a real-to-sim-to-real approach. *CoRL*, 2024.

[15] Z. Chen, A. Walsman, M. Memmel, K. Mo, A. Fang, K. Vemuri, A. Wu, D. Fox, and A. Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *RSS*, 2024.

[16] Z. Jiang, C.-C. Hsu, and Y. Zhu. Ditto: Building digital twins of articulated objects from interaction. In *CVPR*, 2022.

[17] N. Nie, S. Y. Gadre, K. Ehsani, and S. Song. Structure from action: Learning interactions for articulated object 3d structure discovery. *IROS*, 2023.

[18] Z. Mandi, Y. Weng, D. Bauer, and S. Song. Real2code: Reconstruct articulated objects via code generation. *ICLR*, 2025.

[19] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *CoRL*, 2023.

[20] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[21] Y. Yang, F.-Y. Sun, L. Weihs, E. VanderBilt, A. Herrasti, W. Han, J. Wu, N. Haber, R. Krishna, L. Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *CVPR*, 2024.

[22] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *ICML*, 2024.

[23] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *RSS*, 2024.

[24] H. G. Singh, A. Loquercio, C. Sferrazza, J. Wu, H. Qi, P. Abbeel, and J. Malik. Hand-object interaction pretraining from videos. *arXiv preprint arXiv:2409.08273*, 2024.

[25] Y. Chen, C. Wang, Y. Yang, and C. K. Liu. Object-centric dexterous manipulation from human motion data. *CoRL*, 2024.

[26] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. *CoRL*, 2022.

[27] J. Mao, S. Zhao, S. Song, T. Shi, J. Ye, M. Zhang, H. Geng, J. Malik, V. Guizilini, and Y. Wang. Learning from massive human videos for universal humanoid pose control. *arXiv preprint arXiv:2412.14172*, 2024.

[28] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *CoRL*, 2024.

[29] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. *ICLR*, 2024.

[30] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. Learning to act from actionless videos through dense correspondences. *ICLR*, 2024.

[31] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *CoRL*, 2024.

[32] H. Chen, B. Sun, A. Zhang, M. Pollefeys, and S. Leutenegger. Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. *CVPR*, 2025.

[33] C. Yuan, S. Joshi, S. Zhu, H. Su, H. Zhao, and Y. Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation. *arXiv preprint arXiv:2503.18738*, 2025.

[34] L. Y. Chen, C. Xu, K. Dharmarajan, M. Z. Irshad, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *CoRL*, 2024.

[35] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.

[36] M. Lepert, R. Doshi, and J. Bohg. Shadow: Leveraging segmentation masks for cross-embodiment policy transfer. *CoRL*, 2024.

[37] J. Duan, Y. R. Wang, M. Shridhar, D. Fox, and R. Krishna. Ar2-d2: Training a robot without a robot. *CoRL*, 2023.

[38] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *CoRL*, 2024.

[39] E. Teoh, S. Patidar, X. Ma, and S. James. Green screen augmentation enables scene generalisation in robotic manipulation. *arXiv preprint arXiv:2407.07868*, 2024.

[40] B. Chen, H. Jiang, S. Liu, S. Gupta, Y. Li, H. Zhao, and S. Wang. Physgen3d: Crafting a miniature interactive world from a single image. *CVPR*, 2025.

[41] H. Jiang, H.-Y. Hsu, K. Zhang, H.-N. Yu, S. Wang, and Y. Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *arXiv preprint arXiv:2503.17973*, 2025.

[42] Y. Yang, B. Jia, P. Zhi, and S. Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *CVPR*, 2024.

[43] K. Yao, L. Zhang, X. Yan, Y. Zeng, Q. Zhang, L. Xu, W. Yang, J. Gu, and J. Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. *arXiv preprint arXiv:2502.12894*, 2025.

[44] S. Liu, Z. Ren, S. Gupta, and S. Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *ECCV*, 2024.

[45] H.-Y. Hsu, Z.-H. Lin, A. Zhai, H. Xia, and S. Wang. Autovfx: Physically realistic video editing from natural language instructions. *3DV*, 2025.

[46] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[47] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *CVPR*, 2024.

[48] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, M. Anvari, M. Hwang, M. Sharma, A. Aydin, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, S. Savarese, H. Gweon, K. Liu, J. Wu, and L. Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 80–93. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/li23a.html.

[49] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, 2019.

[50] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022.

[51] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun. Depth pro: Sharp monocular metric depth in less than a second. *ICLR*, 2025.

[52] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics*, 2006.

[53] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE T-PAMI*, 1992.

[54] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[55] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. curobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

[56] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE T-RO*, 2023.

[57] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *CVPR*, 2020.

[58] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *CoRL*, 2023.

[59] 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. *IJRR*, 2023.

[60] 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. URL https://arxiv.org/abs/2406.09246.

[61] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[62] S. Belkhale and D. Sadigh. Minivla: A better vla with a smaller footprint. 2024.

[63] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[64] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

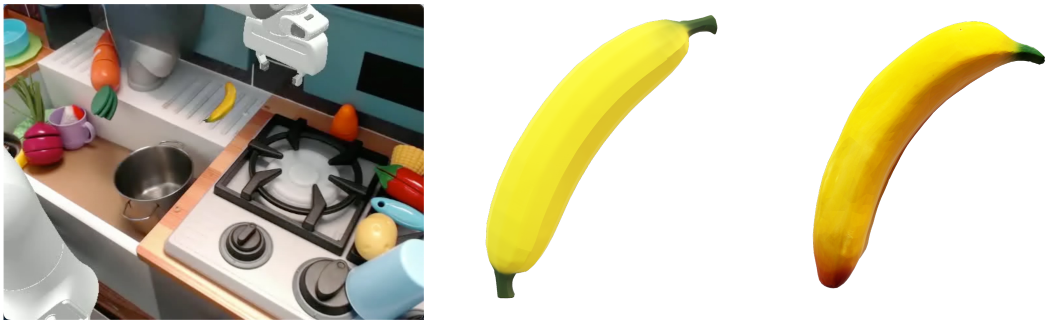
[65] T. Wu, J. Zhang, X. Fu, Y. Wang, L. P. Jiawei Ren, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[66] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

A Experiments

A.1 Physical Scene Recovery

The multiview method is conducted by capturing a short video of the scene using a handheld smartphone, then using Polycam[2] to perform 3D reconstruction from the video. Polycam processes the frames to generate a textured mesh of the environment, which is then used to extract object geometry for simulation. Figure 6 shows the policy learning environment of the task, along with the reconstructed meshes from both the single-view (RoLA) and multiview (Polycam-based) pipelines.



(a) Policy Learning Environment

(b) Single-view Reconstruction Mesh

(c) Multi-view Reconstruction Mesh

Figure 6: Comparison of the policy learning environment and the reconstructed scene meshes from the single-view RoLA pipeline and multiview Polycam pipeline.

A.2 Robotic Data Generation

(1) ACDC[1] is a retrieval-based, single-image data generation method. We follow its original pipeline with modifications to better suit our task. Given a single RGB image, ACDC performs four sequential steps. First, we extract per-object information by prompting GPT-4o[3] to generate captions for all visible objects. These captions are then passed to GroundedSAM[63] to produce object masks. Next, we compute the similarity between each masked object and entries in our object database using SigLip2 [64], retrieving the most visually similar "digital cousin" from OmniObject3D [65]. In the third step, we use metric depth information to align each retrieved mesh with the corresponding object and post-process the matched assets to construct a fully interactive simulated scene. Finally, we employ the same demonstration collection pipeline as RoLA to generate 200 demonstrations per task for imitation policy learning, ensuring fair comparison.

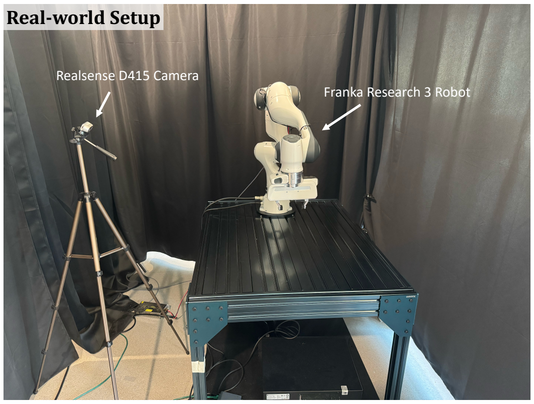


Figure 7: Real-world experiment setup. We employ a Franka Research 3 Robot and a RealSense D415 camera.

(2) RoboEngine [33] is an image-based demonstration augmentation method. We adopt the original codebase without modification. Given a single pre-collected demonstration, RoboEngine segments the robot arm and the manipulated object (i.e., the foreground), then uses a diffusion-based model to synthesize a physically plausible background conditioned on a foreground mask and a scene

[2] https://poly.cam/

[3] https://chatgpt.com/

description. We apply this process to augment each demonstration into 200 demonstrations per task for imitation policy learning, enabling a consistent comparison with other methods.

Figure 8 shows the evaluation tasks (images on the left) and the comparison with the baseline methods. RoLA consistently outperforms the baselines in different tasks.

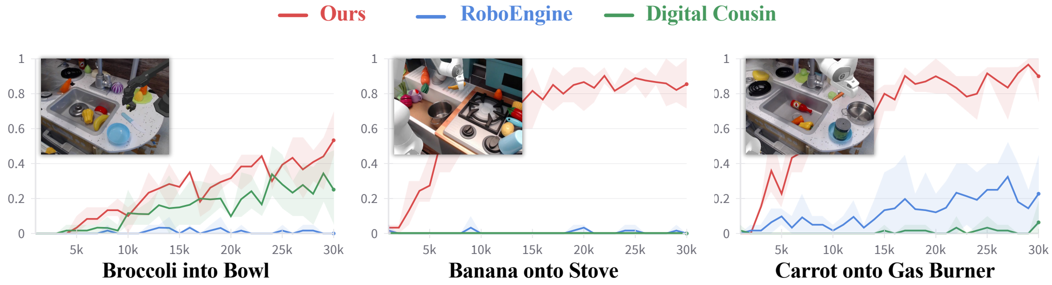


Figure 8: Baseline comparison for robotic data generation.

A.3 Single-Image Robot Learning

Figure 7 illustrates the setup of our real-world experiment. Figure 9 illustrates the comparison between RoLA-generated simulation data and the corresponding sim-to-real deployment in the real world on two tasks: *Manipulation in Cluttered Scenes* and *Pouring Water*. In addition to pick-and-place tasks, we include three dynamic tasks—one of which involves tool use—to demonstrate the versatility of our pipeline. As shown in Figure 10, we present execution trajectories for two simulated tasks: *flip banana* and *push banana*, and one real-world task: *sweep cubes*.

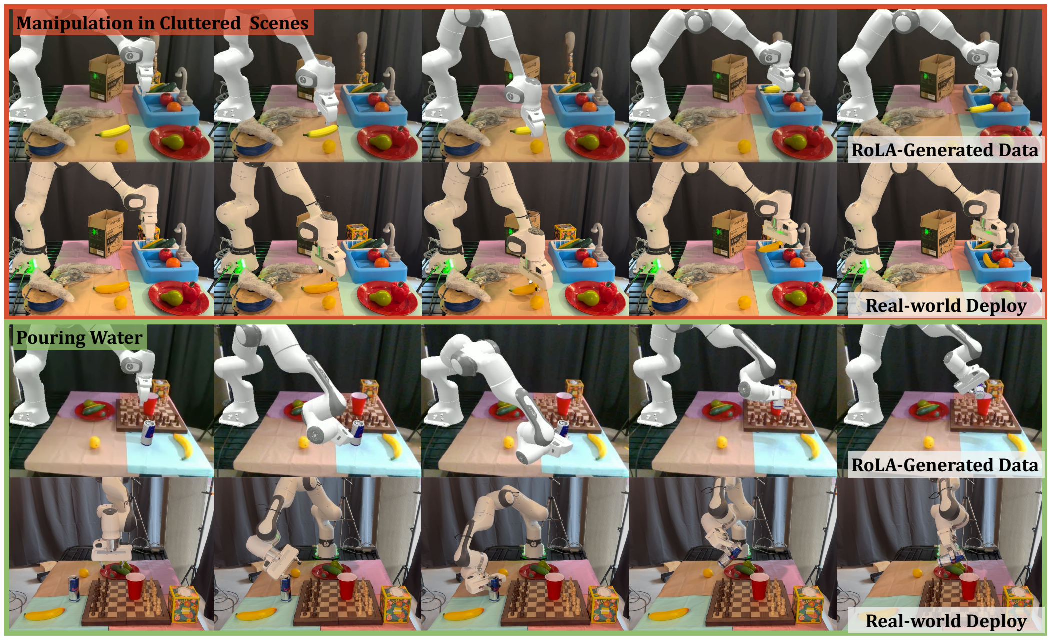


Figure 9: **RoLA-Generated Data vs. Sim2Real Deployment.** Each row pair shows the simulated RoLA-generated scenes (top) and corresponding real-world executions (bottom), demonstrating strong sim-to-real consistency in object placement, robot behavior, and overall task execution.

A.4 VLA Model Training & Evaluation

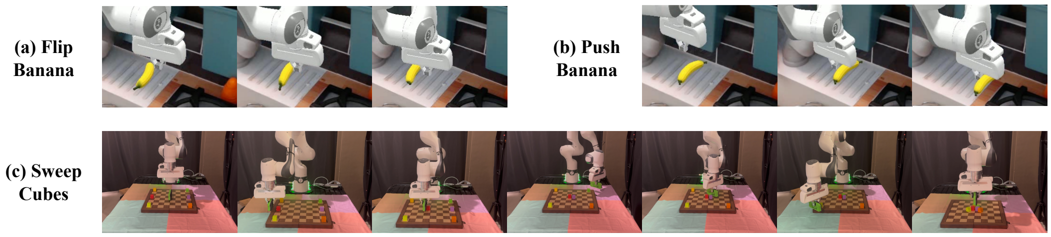


Figure 10: Execution trajectories of dynamic manipulation tasks. (a) Flip Banana and (b) Push Banana are performed in simulation, demonstrating dynamic object interaction. (c) Sweep Cubes is executed in the real world and involves tool use, showcasing the pipeline's ability to generalize to complex and contact-rich tasks.

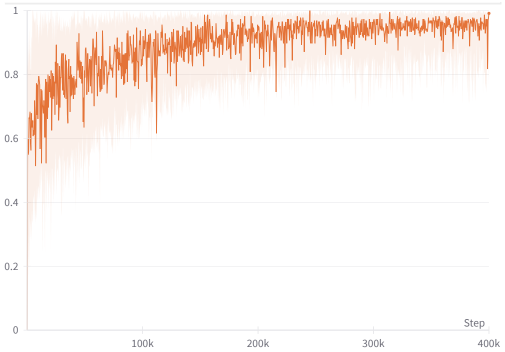


Figure 11: **Training Curve of VLA.** Action token accuracy steadily increases and surpasses 95% within 400K steps.

To demonstrate the scalability of our data generation pipeline, we train a vision-language-action (VLA) model by first capturing real-world photographs and then transforming them into robotic training data. We target the same scale as BridgeData V2 [58]. Specifically, we collect 2,000 real-world images across 12 distinct environments (compared to 24 in BridgeData V2), a process completed within 2 hours of human effort—the only human intervention required. Next, we use GPT-4o to propose approximately 10 tasks per image, yielding around 20,000 unique (image, language instruction) pairs. Our pipeline RoLA is then used to generate approximately 60,000 robotic trajectories, matching the scale of BridgeData V2. Figure 12 displays example images captured in various environments.

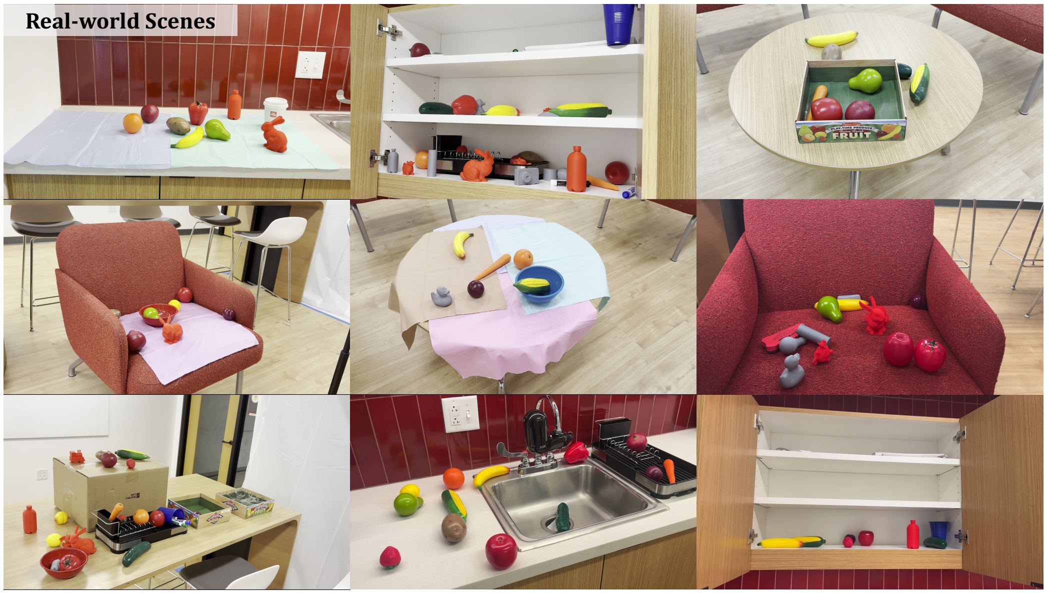


Figure 12: **Example Real-World Scenes.** Sample images captured across diverse environments for vision-language task specification.

For model architecture and training, we strictly follow the MiniVLA [62] setup. The model is trained for 400,000 steps until the action token accuracy exceeds 95%. Training is performed on 8×H100

Language Instruction	Success Rate	Language Instruction	Success Rate
pick up the banana (scene 1)	10/10	pick up the banana (scene 2)	7.5/10

Table 6: Simulation evaluation of our VLA model trained on RoLA-generated data. Each task is tested over 10 trials with randomized object poses, under SimplerEnv [38] setting. Note that *partial success* (score of 0.5) is possible for some tasks, following OpenVLA [60].

GPUs over four days (i.e., 768 GPU-hours). Figure 11 shows the accuracy progression over training steps.

Evaluation. For simulation-based evaluation, we closely follow the SimplerEnv [38] setup, with necessary modifications to the background image, manipulated object, camera pose, and language instruction to align with our task settings. We do not directly evaluate on the original SimplerEnv tasks because our model is not trained on any Open X-Embodiment [66] real-world data, making such a comparison unfair. For the same reason, we do not compare our model against models like OpenVLA [60], which are trained on Open X-Embodiment but not on any of our generated data—such a comparison would be unfair to OpenVLA.

We showcase two additional tasks in Table 6, both using the same language instruction: *pick up the banana*. These tasks are evaluated in distinct background environments to assess the model's ability to generalize. The results show that our model can robustly handle varying visual contexts while correctly executing the instructed behavior.

Below are descriptions of all tasks that requires *partial success* (i.e., a score of 0.5):

1. **Pick up the Banana (Scene 2)**: Partial credit is given when the robot successfully *grasps the banana* but fails to *lift* it.
2. **Put the Green Pepper beside the Lemon**: Partial credit is given when the robot successfully *picks up the green pepper* but does not perform the *place* action.
3. **Take the Banana and Place it beside the Potato**: Partial credit is given when the robot successfully *picks up the banana* but does not perform the *place* action.
4. **Pick up the Red Tomato and Place it beside the Yellow Lemon**: Partial credit is given when the robot successfully *picks up the red tomato* but does not perform the *place* action.
5. **Take the Grey Object beside the Lemon and Place it beside the Yellow Apple**: Partial credit is given when the robot successfully *picks up the grey object* but does not perform the *place* action.

A.5 Learning from Internet Images

We investigate how Internet images can serve as a data source for robot learning, using apple picking as a case study due to the resource constraints. Our motivation is that Internet images exhibit diverse apple types, sizes, environments, and camera viewpoints. If we can generate grasping data from these images, the learned policy could generalize across object variations, different viewpoints, and scenarios. However, due to the huge domain gap between Internet images and real-world robotic environments, direct zero-shot deployment of Internet-learned policy to the real world is infeasible. Instead, we adopt a pretraining–finetuning paradigm: we first pretrain a grasping policy on Internet-generated apple grasping data, then finetune it using a small number of real robot demonstrations for alignment and fast policy adaptation. An illustration of the whole process is shown in Figure 13.

Pretraining. We collect massive Internet apple images, and leverage RoLA to generate robotic data for picking up the apples in images. We filter out those imperfect images and unsuccessful demonstrations, resulting in over 3000 high-quality demonstrations from Internet apple images. Since Internet images are captured from different camera viewpoints and the demonstrations are collected by different robot positions, to unify the data representation, we extract a data sample of an image and the corresponding 6-DoF apple grasp pose in the image frame for each demonstration, which is agnostic to robot positions. Then, we pretrain a neural network that takes an image as input and

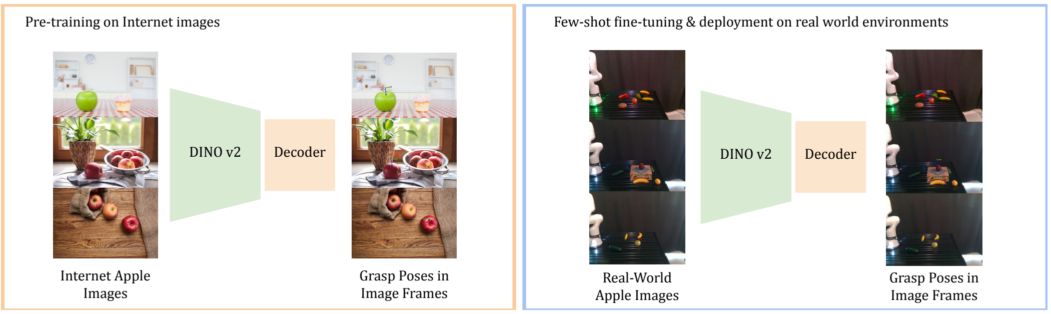

Pre-training on Internet images
DINO v2
Decoder
Internet Apple Images
Grasp Poses in Image Frames
Few-shot fine-tuning & deployment on real world environments
DINO v2
Decoder
Real-World Apple Images
Grasp Poses in Image Frames

Figure 13: The pretraining-finetuning paradigm for learning from Internet images.

outputs the 6-DoF grasp pose in the image frame. The neural network contains a DINO v2 image encoder and a small head for decoding the apple grasp pose. The trained network will serve as an apple grasping prior for the subsequent real robot deployment.

Finetuning. In the real robotic environment, we collect 10/20/50 real robotic demonstrations for picking up an apple, extract the respective camera image and apple grasping pose in the image frame, and leverage these data to fine-tune the pretrained neural network. We set up a baseline by leveraging the same network architecture (DINO v2 + decoder) and directly training the network on the real robotic demonstrations without pretraining on the Internet apple images. During deployment, both the baseline and our network take in the image from the RealScene camera and output the apple grasping pose in the image frame. The grasping pose is then transformed into the robot base frame, and we leverage a motion planner to move the robot gripper to the estimated grasping pose to pick up the apple. We test the real-world grasping success rates and change the apple types, positions, and camera viewpoints during evaluation.

Robotic Demonstrations from Internet Images. Figure 15 shows more examples of robotic demonstrations generated from other Internet images. Our approach is generalizable to different types of objects and images and enables robotic manipulation on in-the-wild images.

B Method

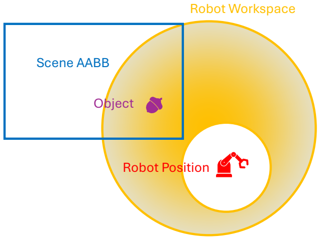


Figure 14: Visualization of the sampling-based method for generating feasible object placements.

B.1 Background Geometry Modeling

Generating background meshes from images is challenging, as removing objects will leave a hole in the background point clouds, and how to in-paint the geometry behind the object remains a problem. One solution is to have another depth estimation on the inpainted background image, but this introduces geometry errors due to the inconsistency of depth estimations. In this paper, we leverage the supported plane to complete the background geometry. Specifically, for pixels in the masked object regions, we shoot a ray from each pixel and check the intersection point of the ray and the supported plane. The intersection points will be used for creating background meshes. We also check the intersections of the ray and the scene's axis-aligned bounding boxes in case the ray does not intersect with the supported plane.

B.2 Robot Placement

For non-robotic images, we propose a sampling-based method to generate feasible placements, as shown in Figure 14.

B.3 Visual Blending

Figure 16 illustrates the detailed visual blending process. We generate foreground masks by comparing the rendered and background depths. For foreground regions, we set the pixels to those in the rendered frames. For background regions, we directly use the background image. This mitigates the rendering artifacts and leads to more photo-realistic results.

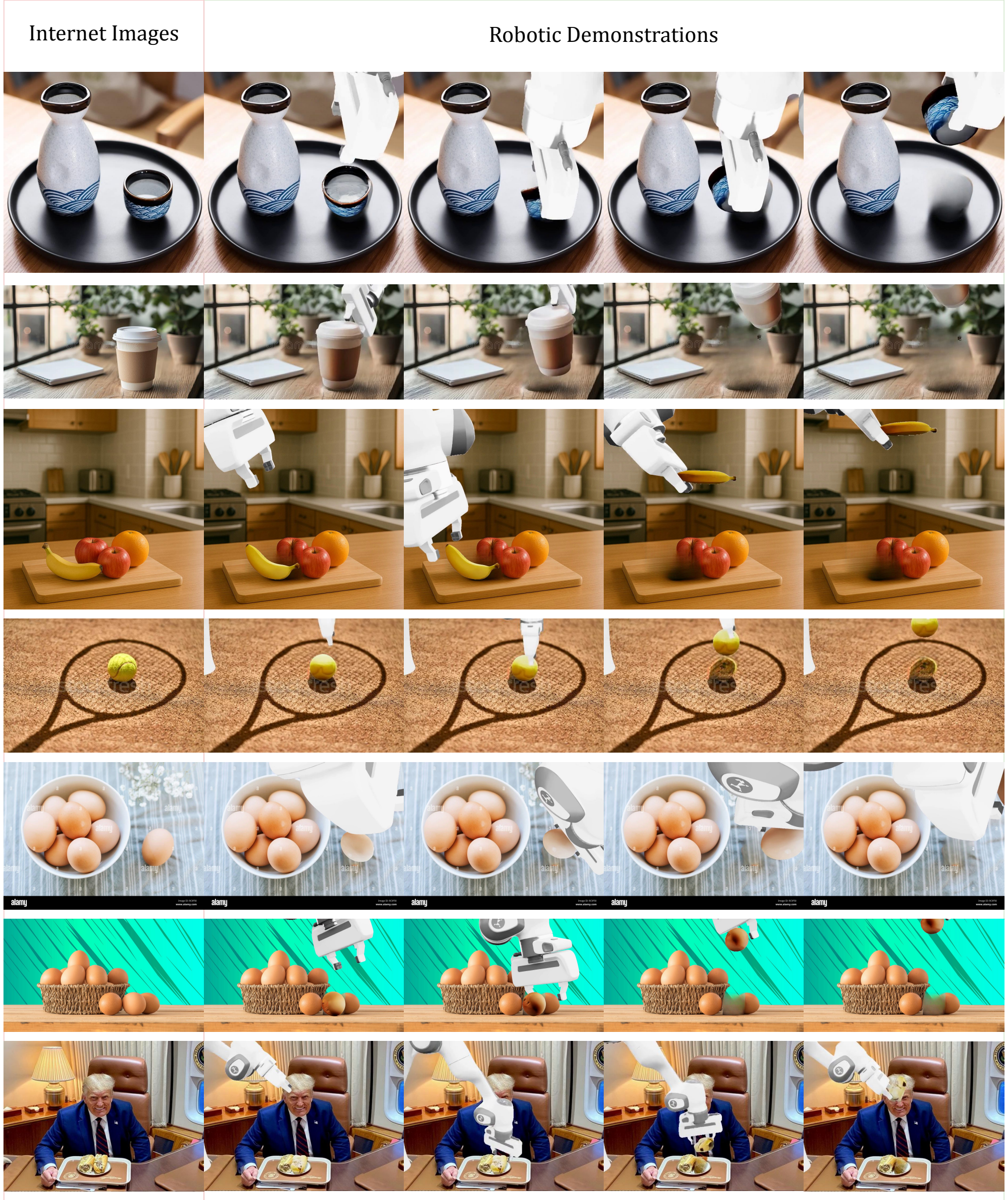


Figure 15: Robotic demonstrations from Internet images.

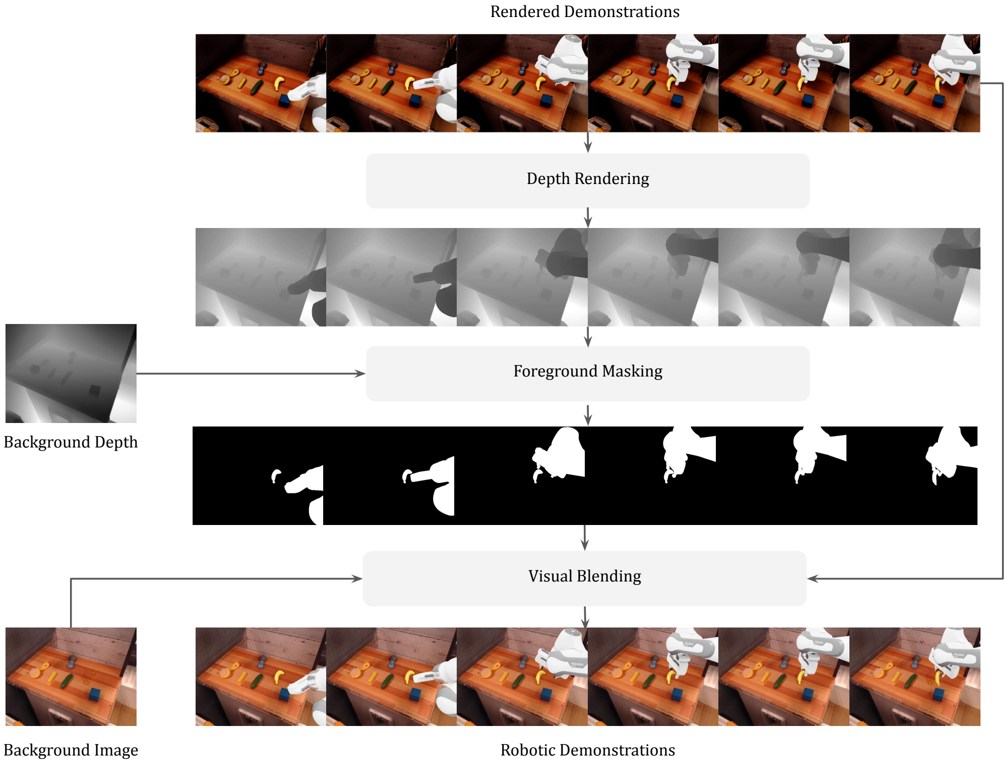


Figure 16: An illustration of visual blending.