

RoboEngine: Plug-and-Play Robot Data Augmentation with Semantic Robot Segmentation and Background Generation

Chengbo Yuan^{*1,3,4}, Suraj Joshi^{*2}, Shaoting Zhu^{*1,3}, Hang Su², Hang Zhao^{1,3,4}, Yang Gao^{†1,3,4}
[RoboEngine.github.io/](https://github.com/RoboEngine)

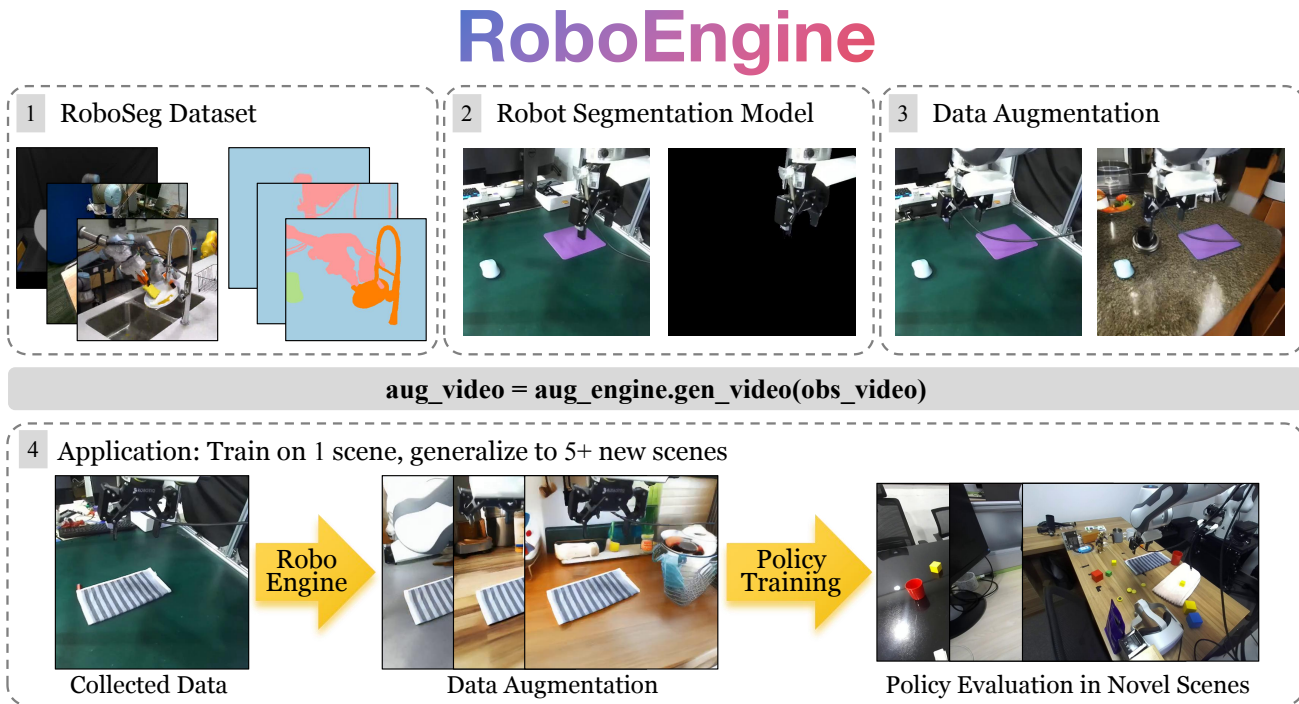


Fig. 1: We introduce RoboEngine, the first plug-and-play toolkit for visual robot data augmentation, enabling physics- and task-aware robot scene generation without any prerequisites. To achieve this, we propose novel datasets, models, and an out-of-the-box toolkit. Extensive real-robot experiments validate the effectiveness of RoboEngine.

Abstract—Visual augmentation has become a crucial technique for enhancing the visual robustness of imitation learning. However, existing methods are often limited by prerequisites such as camera calibration or the need for controlled environments (e.g., green screen setups). In this work, we introduce **RoboEngine**, the first plug-and-play visual robot data augmentation toolkit. For the first time, users can effortlessly generate physics- and task-aware robot scenes with just a few lines of code. To achieve this, we present a novel robot scene segmentation dataset, a generalizable high-quality robot segmentation model, and a fine-tuned background generation model, which together form the core components of the out-of-the-box toolkit. Using **RoboEngine**, we demonstrate the ability to generalize robot manipulation tasks across six entirely new scenes, based solely on demonstrations collected from a single scene, achieving a more than 200% performance improvement compared to the no-augmentation baseline. All datasets, model

weights, and the toolkit will be publicly released.

I. INTRODUCTION

Visuomotor robot manipulation systems have advanced rapidly in recent years [1]. One of the key methods driving this progress is imitation learning [2], [3]. However, imitation learning is widely known to suffer from visual disturbances, particularly when the training environment is limited [4], [5]. While collecting large-scale datasets across diverse scenes can alleviate this issue [6]–[8], this approach is not scalable due to the high costs in terms of time and resources. On the other hand, data augmentation methods [5], [9], [10], particularly generative augmentation techniques, offer a more scalable solution by effortlessly generating diverse visual backgrounds for model training, and have thus gained significant popularity.

Despite their efficiency, existing robot data augmentation methods do not support **plug-and-play usage without prerequisite limitations** (e.g., like the ColorJitter function for computer vision tasks), which restricts their adoption and dis-

¹ Institute for Interdisciplinary Information Sciences, Tsinghua University

² Department of Computer Science and Technology, Tsinghua University

³ Shanghai Qi Zhi Institute

⁴ Shanghai Artificial Intelligence Laboratory

* These authors contributed equally to this work.

† Corresponding at: gaoyangiiis@mail.tsinghua.edu.cn

semination within the community. This limitation arises from a lack of **simultaneous convenience, realism, and diversity**. First, many methods impose pre-requisite conditions for foreground mask generation. For example, GreenAug [5] requires a green screen, and CACTI [9] demands camera calibration. Furthermore, many inpainting-based methods [9], [11] are limited to small object-level augmentations and cannot alter the entire background or layout. Some methods even directly modify the scene using random images or texture [5], which fail to respect physical constraints, leading to degenerated real-world performance due to distribution shifts [12].

To address these limitations, we propose `RoboEngine`, **the first plug-and-play toolkit for physics- and task-aware robot visual data augmentation**. With `RoboEngine`, users can generate physics-constrained backgrounds for robot data **with just a few lines of code**, and without any pre-requisite conditions. We identify that the lack of plug-and-play robot mask generation is a major bottleneck in existing methods [13]. To solve this, we introduce `RoboSeg`, a novel segmentation dataset consisting of 3,800 high-quality annotated robot scene images. By fine-tuning a state-of-the-art (SoTA) model [14] on this dataset, we create the first generalizable “wire-level” robot segmentation model. Additionally, we fine-tune a Diffusion Model [15] on our dataset to enable task-aware background generation. These components are encapsulated into a convenient plug-and-play toolkit, similar to the `ColorJitter` function in the computer vision field*.

To validate the effectiveness of `RoboEngine`, we integrate several SoTA methods [3], [5], [9] into our plug-and-play toolkit, removing all pre-requisite limitations, and conduct real-robot experiments. Training on human demonstrations collected **from only a single scene**, our method enables generalizable manipulation across six entirely new scenes, with significant changes in background, layout, lighting, etc. This results in a 210% and 20% performance improvement over the no-augmentation baseline and the previous SoTA method (our plug-and-play implementation). Additionally, we observe that as the number of augmented demonstrations increases, performance continues to improve to some extent.

In summary, we aim to provide a plug-and-play augmentation toolkit for the robot learning community, akin to the role of `ColorJitter` in computer vision. Our main contributions are as follows:

- We introduce a novel dataset, `RoboSeg`, which contains 3,800 high-quality segmentation images of robot scenes. Based on this dataset, we train the first generalizable high-quality robot segmentation model.
- We integrate our segmentation model and a new fine-tuned physics- and task-aware Diffusion model into the plug-and-play `RoboEngine` toolkit. Users can achieve data augmentation with just a few lines of code, without any pre-requisite limitations.
- We conduct extensive real-robot experiments to validate the effectiveness of `RoboEngine`. Training on

demonstrations from only one scene, our method enables generalizable manipulation across six entirely new scenes, achieving significant improvements over previous methods.

II. RELATED WORKS

A. Visual Generalizable Imitation Learning

Imitation Learning (IL) [2], [16]–[18] has become a key approach for visuomotor robot control [1], [19], where deep neural networks are used to predict motor actions directly from visual inputs. In this paper, we adopt Diffusion Policy (DP) [3], a state-of-the-art IL method, as our robot policy architecture. However, raw DP is sensitive to visual disturbances [4], [5]. To improve visual robustness and generalization in IL, several strategies have been proposed, including novel architectures [20]–[22], vision pretraining [23], [24], and enhanced data collection and synthesis approaches [8], [9], [25]. Our work focuses on addressing these challenges through data augmentation [5] using a plug-and-play robot data synthesis engine.

B. Robot Data Augmentation

Collecting robot data is both costly and labor-intensive [26], especially in diverse environments [7]. To mitigate this, various data augmentation techniques [5], [9]–[11], [13], [25], [27]–[29] have been proposed, enabling the synthesis of diverse robot data at minimal cost. Traditional methods, such as Cropping and `ColorJitter`, are effective for in-domain visual generalization [3], but struggle when handling large visual changes. Generative methods [5] can better manage such changes, but they are not plug-and-play due to their pre-requisite requirements [25]. For example, [9], [11] necessitates camera calibration. It also depends on scene objects for inpainting, which can lead to unpredictable behavior unless hyperparameters are carefully adjusted for each scene. Additionally, these methods [9]–[11], [27] offer limited visual variation, which may fail under extreme conditions. While `GreenAug` [5] can generate more significant visual changes, it requires a green screen setup and struggles in unstructured environments [7]. Moreover, it neglects the physical reliability of the environment, which can lead to out-of-distribution issues [12].

III. METHOD

In this section, we provide details of our plug-and-play `RoboEngine` augmentation methods. To generate robot masks without any pre-requisites, we introduce a new robot scene segmentation dataset and develop the first generalizable high-quality robot segmentation model. We then fine-tune a Diffusion Model [15] and integrate both models into a seamless plug-and-play pipeline.

A. Task Definition

Given a robot manipulation demonstration dataset $\mathcal{D} = I_1, J_1, I_2, J_2, \dots, I_n, J_n$, where I_i denotes image data and J_i represents associated information such as language instructions, robot proprioception, and actions, `RoboEngine` aims

*pytorch.org/vision/main/generated/torchvision.transforms.ColorJitter.html

to augment \mathcal{D} into \mathcal{D}_a using a trained segmentation model and a generative model.

For each image $I_i = \{R_i, O_i, B_i\}$ in \mathcal{D} , where R_i is the robot arm area, O_i is the task-related objects area (e.g., for “Put Mouse to the Pad,” O_i includes the mouse and pad), and B_i is the background, we first apply semantic segmentation to split the areas. We then generate a new background B_i^* . Theoretically, the closer the distribution of B_i^* is to the robot’s deployment environment, the better the robot’s manipulation performance [12]. Finally, we combine R_i , O_i , and B_i^* to create the augmented image $I_i^* = \{R_i, O_i, B_i^*\}$. The resulting augmented dataset is represented as $\mathcal{D}_a = \{I_1^*, J_1, I_2^*, J_2, \dots, I_n^*, J_n\}$.

B. RoboSeg Dataset and Robot Segmentation Model

As noted in previous work [13], we observe that existing off-the-shelf segmentation models, such as the Segment Anything Model V2 (SAM-V2) [30] and LISA [31], struggle to segment robots with high accuracy. This limitation often necessitates prerequisites such as a green screen setup [5] or camera calibration [9], [11] for generating accurate robot masks.

To address this, we propose RoboSeg, a novel dataset with high-quality robot scene segmentation annotations. RoboSeg contains 3,800 images randomly selected from over 35 robot datasets [3], [6], [7], [32]–[34], covering a broad range of robot types (e.g., Franka, WindowX, HelloRobot, UR5, Sawyer, Xarm, etc.), camera views, and background environments. A visualization of RoboSeg can be found in Figure 2. For each image, we provide three types of masks: (1) “robot-main”: the part of the robot arm connected to the robot gripper in pixel space, (2) “robot-auxiliary”: other parts of the robot, such as the robot base in some images, (3) “object”: the area of all task-related objects. Note that even the robot’s wires are annotated, creating **extremely fine-grained masks**. Additionally, each image includes the task instruction, and we use GPT-4o [35] to generate 10 brief descriptions for each scene.

Based on RoboSeg, we fine-tune the state-of-the-art (SoTA) language-conditioned segmentation model EVF-SAM [14] to create a new robot segmentation model, Robo-SAM, which achieves high-quality open-world robot segmentation. We choose the language-conditioned version rather than the original SAM [30] because we find that language provides additional visual cues that enhance the base model’s performance. Details on model training can be found in Appendix V-A.

C. Plug-and-Play RoboEngine Augmentation

Given a robot scene image, we first generate the robot mask and task-related object masks using Robo-SAM and EVF-SAM [14] (conditioned on the object name in the task instruction). We then use a generative model to create a physics- and task-aware background based on the previously generated masks. For this, we use BackGround-Diffusion [15], which generates a foreground-aware background with physical constraints given a foreground mask and scene description. We fine-tune this model on our RoboSeg dataset to eliminate unreasonable generations in some cases. More details can be

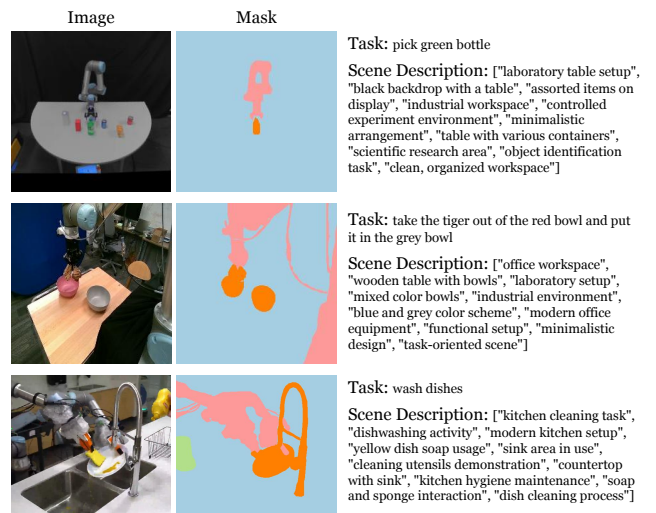


Fig. 2: Our RoboSeg dataset provides high-quality, fine-grained semantic segmentation annotations, covering a wide diversity of robots and environments.

found in Appendix V-A. For scene descriptions, we collect all descriptions from RoboSeg to create a prompt pool, from which we randomly select one during usage to generate a randomly new scenes.

We integrate our segmentation and background generation models into a plug-and-play toolkit, RoboEngine, which requires no pre-requisite setup. Users can generate augmented robot data with just a few lines of code:

```

1 from robo_engine import RoboEngine
2 # aug_method: engine, texture, imagenet.....
3 aug_method = 'robo_engine'
4 engine = RoboEngine(
5     robo_seg_method=['robo_sam_video'],
6     obj_seg_method=['evf_sam_video'],
7     aug_method=aug_method,
8     batch_size=32
9 )
10
11 aug_video = aug_engine.gen_video(obs_video)

```

This is similar to the usage of fundamental traditional augmentation methods, such as ColorJitter or RandomCrop, in the computer vision community. We hope this will facilitate the adoption and wider use of generative visual robot data augmentation methods.

IV. EXPERIMENTS

A. Segmentation Results

Experiment Setting. We first evaluate our new robot segmentation model, Robo-SAM, against other off-the-shelf state-of-the-art (SoTA) segmentation models. We evaluate all models on two datasets: (1) “Test Set”, which contains 97 images of novel scenes extracted from the same source sub-dataset as our proposed RoboSeg dataset, and (2) “Zero-shot Set”, which consists of 45 robot manipulation images randomly downloaded from the internet. We compare our model with three widely used language-conditioned SoTA segmentation models: ClipSeg [36], LISA [31], and EVF-SAM [14], all using the “robot” instruction prompt. Generalized Intersection

TABLE I: Segmentation Performance Comparison of different methods on the Test Set and Zero-shot Set, evaluated using the Generalized IoU (GIoU) metric [37].

Method	Test Set \uparrow	Zero-shot Set \uparrow
CLIPSeg [38]	0.2810	0.4049
LISA [31]	0.6040	0.7571
EVF-SAM [14]	0.6290	0.7777
Robo-SAM (Ours)	0.8620	0.9037

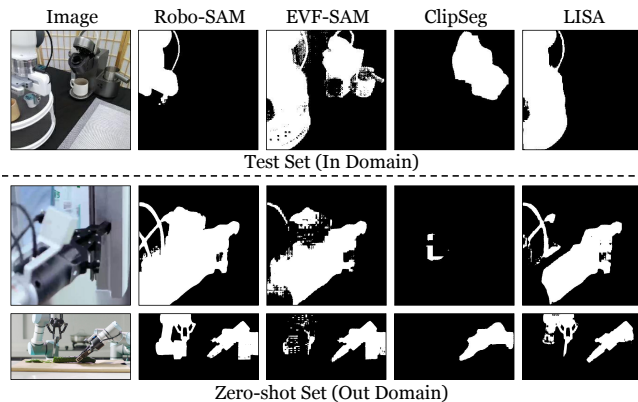


Fig. 3: Comparison of Segmentation Results between our new Robo-SAM model and other baselines. Only Robo-SAM produces usable segmentation masks for downstream augmentation applications.

over Union (GIoU) [37] is used as the evaluation metric for our quantitative experiments.

Results. The quantitative results are shown in Table I. Our new Robo-SAM model outperforms all other baselines with a substantial improvement (> 0.12 GIoU) on both the “Test” and “Zero-shot” sets. We also visualize the predictions of all models in Figure 3. As shown, none of the baselines achieve usable robot segmentation, as observed in previous work [13]. In contrast, our Robo-SAM model aligns well with the robot shape, even in challenging novel scenes. This opens new possibilities for calibration-free and plug-and-play robot segmentation and augmentation techniques.

B. Augmentation Baseline and Results

We next compare our proposed RoboEngine augmentation with previous augmentation methods. For convenience, we have encapsulated all of these methods into our toolkit, allowing users to easily switch between them by adjusting a configuration parameter (“aug_method”). Note that, for calibration-free and plug-and-play usage, all of these methods **depend on the Robo-SAM segmentation** model we proposed earlier. The baselines we compare with are as follows:

- **Background** [5]: generates a scene image using a generative model from a scene description (which does not consider physics feasibility). The description is randomly selected from the description pool.
- **Inpainting** [9], [11]: uses SAM-V2 [30] to obtain masks for all objects, then inpaints 5 task-irrelevant objects.

- **ImageNet** [13]: randomly selects an image from the ImageNet [39] training set as the background.
- **Texture** [5]: randomly selects a texture as the background.
- **No aug** [3], [8]: no generative augmentation.

For the “Background” method, we use Stable Diffusion V2.1[†] as the generative model. For the “Inpainting” method, we use Stable Inpainting[‡]. For the “Texture” method, we select textures from the MIL-Dataset [40][§]. A visualization of all methods is shown in Figure 4 (a). It can be seen that RoboEngine is the only method that provides both high diversity and physics feasibility, aligning most closely with robot deployment environments.

C. Real Robot Setting

Next, we train real robot policy on these augmented method to evaluate the effectiveness of all augmentation methods.

Robot and Policy Training. We follow the hardware setup of the DROID platform [7], using a Franka Panda robot arm with a Robotiq gripper and absolute pose control, but only with one third-view RGB image stream. For policy training, we adopt the Diffusion Policy (DP) [3], a state-of-the-art algorithm for robot learning. We make the following modifications: (1) Replace the original image encoder in DP with a DINOv2-Base [41] encoder to improve visual understanding [8], [16]. (2) Set the observation horizon to 1 to better suit the current augmentation setup. We also set the action horizon to prediction horizon, as we find it performs better with this setting [22]. (3) Train all tasks for 1000 epochs to ensure model convergence. For other hyperparameters, we follow the guidelines in [8]. Traditional image augmentation techniques, such as ColorJitter and RandomCrop, are applied across all methods during DP training [8].

Tasks and Metric. We evaluate using two tasks: “Fold Towel” and “Put Mouse on the Pad”. These tasks cover long-horizon (multi-stage), deformable (folding towel), and precise (grasping mouse) manipulation. Since sparse success rates (SR) may obscure performance differences, we also record the behavior score for each task. Detailed criteria for SR and the score can be found in Appendix V-B. For clarity, we normalize the behavior score by dividing the original score by the maximum possible score in Table II, resulting in a normalized maximum score of 1. Both tasks are multi-stage (grasping and manipulation stages), so we record SR and scores for both grasping and task completion.

For the “Fold Towel” task, we collect 50 demonstrations, with towels placed in a 35cm \times 35cm grid. For the “Put Mouse” task, we collect 100 demonstrations, with the mouse and pad placed in a 15cm \times 15cm grid. By default, we apply each augmentation method to the demonstrations once, so the augmented dataset still contains 50 and 100 demos. Data is only collected in one scene, and we test the two tasks in 4 and 2 completely new scenes, respectively. The scene visualizations can be found in Figure 4 (b). Note that our project focuses

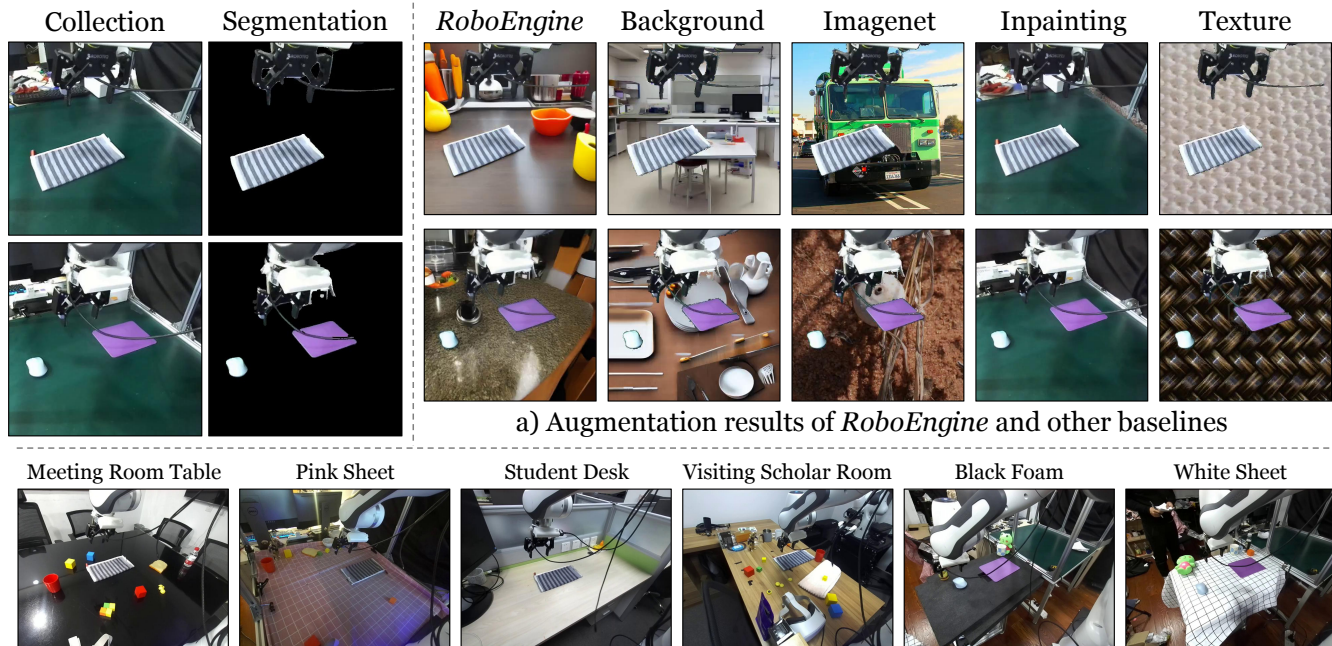
[†]<https://huggingface.co/stabilityai/stable-diffusion-2-1>

[‡]<https://huggingface.co/docs/diffusers/using-diffusers/inpaint>

[§]https://huggingface.co/datasets/eugeneteoh/mil_data

TABLE II: Comparison of Different Augmentation Methods on real robot performance, using both behavior score and success rate as metrics. All data are collected from a single scene, and the evaluation is conducted across 6 completely new scenes. See Figure 4 (b) for scene visualization.

	Fold Towel (Good Grasp) \uparrow	Fold Towel (Finish) \uparrow	Put Mouse (Good Grasp) \uparrow	Put Mouse (Finish) \uparrow	Average \uparrow
No aug	0.36 / 40.6%	0.29 / 15.6%	0.15 / 6.0%	0.07 / 0.0%	0.20 / 15.6%
Inpainting	0.36 / 40.6%	0.34 / 34.4%	0.63 / 12.5%	0.10 / 0.0%	0.24 / 21.8%
Background	0.50 / 53.1%	0.54 / 62.5%	0.46 / 37.5%	0.32 / 18.8%	0.45 / 43.0%
ImageNet	0.50 / 53.1%	0.52 / 62.5%	0.56 / 43.7%	0.39 / 18.8%	0.48 / 44.5%
Texture	0.50 / 50.0%	0.54 / 62.5%	0.63 / 56.2%	0.44 / 25.0%	0.51 / 48.4%
RoboEngine	0.56 / 56.2%	0.59 / 68.7%	0.79 / 75.0%	0.58 / 43.7%	0.62 / 60.9%



b) Different out-of-domain environment for policy evaluation

Fig. 4: (a) Augmentation results using different methods. *RoboEngine* is the only method that simultaneously satisfies both physics constraints and high visual diversity. (b) Visualization of real robot evaluation environment. All scenes exhibit significant visual differences from the scene used for data collection.

TABLE III: Augmentation speed comparison across methods. The time (in seconds) required for augmenting a single frame is measured for each method.

Inpainting	Background	Imagenet	Texture	RoboEngine
3.90 sec	1.91 sec	0.97 sec	0.97 sec	2.17 sec

on solving visual generalization, not spatial generalization, so tables in all scenes share a similar height range of 73 to 76 cm.

D. Policy Evaluation Results

We report the average score and success rate for all tasks in Table II, with per-scene results available in Appendix V-C. Additionally, the time cost for all methods in the “Fold Towel” task is provided in Table III (with batch size set to 1 for measurement). Our method demonstrates a significant improvement, outperforming the best baselines by 20%, and achieves a 210% performance boost compared to the no-augmentation baseline.

For the “Inpainting” method [9], [11], we observe that performance heavily depends on the task-irrelevant object masks and is very sensitive to hyperparameters (e.g., the number of inpainting objects and their sizes). This often results in small visual changes, which are insufficient for robust visual generalization. Additionally, the time cost for this method is the highest, so we do not recommend it for practical use.

For the “Background”, “ImageNet”, and “Texture” methods, all show substantial performance gains compared to “No aug”, with “Texture” outperforming the others slightly (consistent with findings in [5]). Although the performance is lower than *RoboEngine*, both “Texture” and “ImageNet” are faster, as they do not require running a foundation generative model like Stable Diffusion [15]. We recommend using *RoboEngine* when computational resources are available for optimal performance, and opting for “Texture” or “ImageNet” when resources are limited.

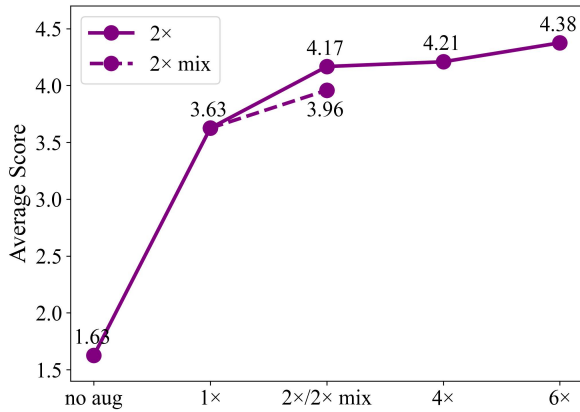


Fig. 5: Performance scaling trend experiment results. We report the average behavior score (un-normalized) for the “Fold Towel (Finish)” task across 3 novel scenes. Raw results can be found in Appendix V-D.

E. Performance Scaling Trend

We conduct an experiment to verify whether increasing the number of augmented demonstrations leads to better performance. For efficiency, we focus on the “Fold Towel (Finish)” task and evaluate the policy in 3 novel scenes (Meeting Room Table, Pink Sheet, Student Desk). We use our proposed `RoboEngine` augmentation method, which performs best among all methods (as shown in Table II). We represent the original augmented dataset as “1x” (50 demos), and “ N x” represents datasets enlarged by a factor of N (e.g., $50 \times N$ demos). We also investigate whether mixing the original data with augmented data benefits policy training, using a “2x mix” variant, which includes 50 original demos and 50 augmented demos. For a fair comparison, we adjust the number of training epochs for each variant to ensure the training steps are similar [8]. More details can be found in Appendix V-D.

The results, shown in Figure 5, reveal two key findings: (1) Mixing the original dataset with augmented data does help to improve performance (“2x mix” vs. “1x”), but show no advantage to only using augmented data (“2x mix” vs. “2x”). (2) Increasing the number of augmented demonstrations does lead to performance improvements, but the rate of improvement gradually slows down and eventually reaches a bottleneck.

V. CONCLUSION

Conclusion. In this work, we introduce `RoboEngine`, the first plug-and-play visual robot data augmentation toolkit that enables physics- and task-aware robot scene background generation with no pre-requirements. By fine-tuning existing state-of-the-art models on our newly proposed `RoboSeg` dataset, we create the first high-quality, generalizable robot segmentation model, `Robo-SAM`, along with a new physics- and task-aware background generation Diffusion model. We encapsulate the entire pipeline into an out-of-the-box toolkit, allowing users to perform augmentation with just a few lines of code. Extensive real-robot experiments demonstrate the

effectiveness of `RoboEngine` augmentation. We hope our work will help alleviate visual robustness issues in imitation learning and serve as a standard augmentation toolkit for the robot learning community.

Limitation. While `RoboEngine` provides significant convenience for visual robot data augmentation, it has some limitations. First, our method does not handle temporal consistency between frames, which could be addressed by recent video diffusion models [42]. Second, we do not consider multi-view or 3D augmentation. This limitation may be overcome by using depth estimation techniques [43] to lift the generated background into 3D, followed by re-rendering.

APPENDIX

A. Details of Model Training

Robo-SAM. We choose EVF-SAM [14] as our base model. EVF-SAM has two versions: the original version for instance segmentation and the “multitask” version for semantic segmentation[¶]. We find that the latter version significantly outperforms the former, therefore, we choose the “multitask” version as our base model. For fine-tuning, we use 3562 images as the training set and 92 images as the validation set. All other images from `RoboSeg` are used for model evaluation. We fine-tune the model for 30 epochs, with a learning rate of $1e-5$ and a batch size of 32.

BackGround-Diffusion. We fine-tune `BackGround-Diffusion` [15] on the `RoboSeg` dataset for 100 epochs, with a learning rate of $5e-3$ and a batch size of 32. The scene descriptions of the images are generated by GPT-4o [35]. All images in `RoboSeg` are used for fine-tuning.

B. Evaluation Criteria

The criteria for the behavior score are as follows:

Fold Towel (Grasp Stage).

- 0: The gripper does not contact the towel.
- 1: The gripper grasps the towel, but not at the left edge.
- 2: The gripper grasps at the left edge, but is outside a 7.5 cm range around the center of the edge.
- 3: The grasping is within the 7.5 cm perfect area of the left edge.

Fold Towel (Fold Stage).

- 0: Not folded.
- 1: The overlay is smaller than 1/3 of the maximum overlay.
- 2: The overlay is larger than 1/3 but smaller than 2/3.
- 3: The overlay is larger than 2/3 of the maximum overlay.

Put Mouse (Grasp Stage).

- 0: No contact with the mouse.
- 1: Contact with the mouse, but not grasped stably, or dropped during lifting.
- 2: The mouse is grasped stably, but has moved more than 3 cm on the table before grasping.
- 3: The mouse is grasped stably with small movement on the table.

Put Mouse (Put Stage).

- 0: The mouse is dropped during the put, and falls off the pad.

[¶]<https://github.com/hustvl/EVF-SAM>

TABLE IV: The raw evaluation results for all methods are presented, with the behavior score as the metric (un-normalized score). The success rate can be calculated based on the criteria outlined in Appendix V-B.

	Fold Towel (Grasp) ↑				Fold Towel (Finish) ↑				Put Mouse (Grasp) ↑		Put Mouse (Finish) ↑	
	Student Desk	Pink Sheet	Meeting Room	Scholar Room	Student Desk	Pink Sheet	Meeting Room	Scholar Room	Black Boam	White Sheet	Black Boam	White Sheet
No aug	0.875	1.125	1.25	1.125	1.625	1.625	1.625	2	0.875	0	0.875	0
Inpainting	0.625	1.625	0.875	1.25	1.375	3	1.75	2.125	1.25	0	1.25	0
Background	1.875	1.25	1.5	1.375	4.125	2.875	3	2.875	1.625	1.125	2.375	1.5
ImageNet	1.75	1.5	1.25	1.5	3.75	2.75	2.875	3.125	1	2.375	1.25	3.375
Texture	1.875	1.5	1.5	1.125	4.125	3.5	3	2.25	1.375	2.375	2.125	3.125
RoboEngine	2	1.75	1.5	1.5	4.5	3.25	3.125	3.375	2.5	2.25	3.875	3.125

TABLE V: Raw results of the scaling trend experiment, conducted on the 'Fold Towel (Finish)' task, with the behavior score (un-normalized) as the metric.

	Pink Table ↑	Meeting Room ↑	Student Table ↑
2 × mix	4.375	2.875	4.625
No aug	1.625	1.625	1.625
1 ×	3.25	3.125	4.5
2 ×	4.5	2.875	5.125
4 ×	5	3	4.625
6 ×	4.625	3.375	5.125

- 1: The mouse is dropped during the put, but falls on the pad.
- 2: The mouse is put on the pad stably, but outside the 10 cm × 10 cm center area.
- 3: The mouse is put in the center area stably.

For each stage, we consider it a success execution if the behavior score is greater than or equal to 2.

C. Raw Evaluation Results

For each task, we perform 8 evaluations per scene and compute the average behavior score. The raw evaluation results are provided in Table IV.

D. Details of Scaling Trend Experiment

We train all variants with the following settings: (1) "1 ×" for 1000 epochs, (2) "2 ×" and "2 × mix" for 700 epochs, (3) "4 ×" for 400 epochs, and (4) "6 ×" for 300 epochs. For resource efficiency, we only conduct experiments on the "Fold Towel (Finish)" task. For each policy and scene combination, we perform 8 rollouts and record the average behavior score. The raw results can be found in Table V.

ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (2022ZD0161700), National Natural Science Foundation of China (62176135), Shanghai Qi Zhi Institute Innovation Program SQZ202306 and the Tsinghua University Dushi Program.

REFERENCES

- [1] M. A. Goodale, "Visuomotor control: Where does vision end and action begin?" *Current Biology*, vol. 8, no. 14, pp. R489–R491, 1998.
- [2] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, "Aloha unleashed: A simple recipe for robot dexterity," *arXiv preprint arXiv:2410.13126*, 2024.
- [3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.
- [4] A. Xie, L. Lee, T. Xiao, and C. Finn, "Decomposing the generalization gap in imitation learning for visual robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3153–3160.
- [5] E. Teoh, S. Patidar, X. Ma, and S. James, "Green screen augmentation enables scene generalisation in robotic manipulation," *arXiv preprint arXiv:2407.07868*, 2024.
- [6] O. X.-E. Collaboration, "Open X-Embodiment: Robotic learning datasets and RT-X models," <https://arxiv.org/abs/2310.08864>, 2023.
- [7] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.
- [8] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao, "Data scaling laws in imitation learning for robotic manipulation," *arXiv preprint arXiv:2410.18647*, 2024.
- [9] Z. Chen, Z. Mandi, H. Bharadhwaj, M. Sharma, S. Song, A. Gupta, and V. Kumar, "Semantically controllable augmentations for generalizable robot learning," *The International Journal of Robotics Research*, p. 02783649241273686, 2024.
- [10] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter, *et al.*, "Scaling robot learning with semantically imagined experience," *arXiv preprint arXiv:2302.11550*, 2023.
- [11] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar, "Cacti: A framework for scalable multi-task multi-scene visual imitation learning," *arXiv preprint arXiv:2212.05711*, 2022.
- [12] L. Fan, K. Chen, D. Krishnan, D. Katabi, P. Isola, and Y. Tian, "Scaling laws of synthetic images for model training... for now," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7382–7392.
- [13] 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," *arXiv preprint arXiv:2409.03403*, 2024.
- [14] Y. Zhang, T. Cheng, R. Hu, L. Liu, H. Liu, L. Ran, X. Chen, W. Liu, and X. Wang, "Evl-sam: Early vision-language fusion for text-prompted segment anything model," *arXiv preprint arXiv:2406.20076*, 2024.
- [15] A. E. Eshratifar, J. V. Soares, K. Thadani, S. Mishra, M. Kuznetsov, Y.-N. Ku, and P. De Juan, "Salient object-aware background generation using text-guided diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7489–7499.
- [16] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.
- [17] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.
- [18] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.
- [19] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.
- [20] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy," *arXiv preprint arXiv:2403.03954*, 2024.
- [21] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu, "Generalizable humanoid manipulation with improved 3d diffusion policies," *arXiv preprint arXiv:2410.10803*, 2024.

- [22] C. Wang, H. Fang, H.-S. Fang, and C. Lu, "Rise: 3d perception makes real-world robot imitation simple and effective," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 2870–2877.
- [23] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.
- [24] G. Jiang, Y. Sun, T. Huang, H. Li, Y. Liang, and H. Xu, "Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets," *arXiv preprint arXiv:2410.22325*, 2024.
- [25] J. Wang, Y. Qin, K. Kuang, Y. Korkmaz, A. Gurumoorthy, H. Su, and X. Wang, "Cyberdemo: Augmenting simulated human demonstration for real-world dexterous manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17952–17963.
- [26] A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.
- [27] Z. Zhuang, R. Wang, N. Ingelhart, V. Kyrki, and D. Kragic, "Enhancing visual domain robustness in behaviour cloning via saliency-guided augmentation," in *CoRL Workshop on Safe and Robust Robot Learning for Operation in the Real World*, 2024.
- [28] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Boloni, "Pay attention!-robustifying a deep visuomotor policy through task-focused visual attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4254–4262.
- [29] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter, *et al.*, "Scaling robot learning with semantically imagined experience," *arXiv preprint arXiv:2302.11550*, 2023.
- [30] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.
- [31] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "Lisa: Reasoning segmentation via large language model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9579–9589.
- [32] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li, "Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation," *arXiv preprint arXiv:2402.15487*, 2024.
- [33] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.
- [34] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," *arXiv preprint arXiv:2410.07864*, 2024.
- [35] 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.
- [36] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 7086–7096.
- [37] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.
- [38] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 7086–7096.
- [39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [40] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Conference on robot learning*. PMLR, 2017, pp. 357–368.
- [41] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.
- [42] Z. Zhang, B. Wu, X. Wang, Y. Luo, L. Zhang, Y. Zhao, P. Vajda, D. Metaxas, and L. Yu, "Avid: Any-length video inpainting with diffusion model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7162–7172.
- [43] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang, "Video depth anything: Consistent depth estimation for super-long videos," *arXiv preprint arXiv:2501.12375*, 2025.