RehearseVLA: Simulated Post-Training for VLAs with Physically-Consistent World Model

Supplementary Material

1. Algorithm

We show the full reinforcement learning post-training algorithm in Algorithm 1.

2. Comparison to Octo.

We compare with Octo [5] in Table 1. As shown in the table, our method outperforms Octo across all four LIBERO suites despite using only 10% training data, manifesting its effectiveness.

Method	LIBERO-Goal	LIBERO-Object	LIBERO-Spatial	LIBERO-Long
Octo	84.6	85.7	78.9	51.1
Ours	**86.4**	**86.6**	**87.6**	**57.8**

Table 1. **Quantitative comparison with Octo on LIBERO.**

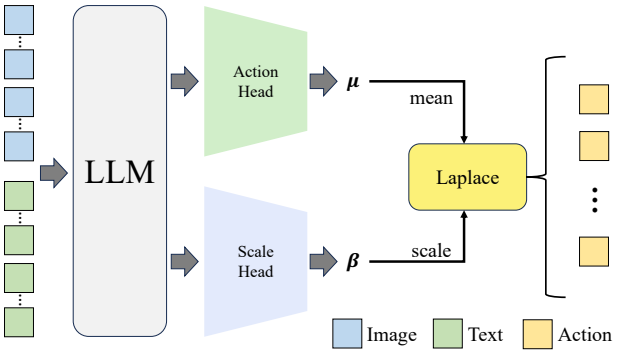


Figure 1. **Architecture for uncertainty-aware action generation.** The deterministic action output of the VLA policy is augmented with a parallel Laplace scale head to model action uncertainty.

3. More Implementation Details

3.1. Deatils of Scale Head

Our method builds upon OpenVLA-OFT [3], which predicts continuous actions via an action head that takes hidden states $f \in \mathbb{R}^d$ as input and employs L1 loss for action regression:

$$\mathcal{L}_{\text{L1}} = \|\mathbf{a}_{\text{gt}} - \boldsymbol{\mu}\|_1 \quad \text{where } \boldsymbol{\mu} = \text{MLP}_{\text{action}}(f). \tag{1}$$

To model heteroscedastic uncertainty in action prediction, we introduce a scale head with the same MLP architecture as the action head, as shown in Figure 1. This scale head outputs log-scale parameters $\boldsymbol{\beta}$ through:

$$\boldsymbol{\beta} = \text{MLP}_{\text{scale}}(h), \tag{2}$$

and is trained with negative log-likelihood (NLL) loss under a Laplace distribution assumption:

$$\mathcal{L}_{\text{NLL}} = \underbrace{|\mathbf{a}_{\text{gt}} - \boldsymbol{\mu}| \cdot e^{-\boldsymbol{\beta}}}_{\text{Data fit}} + \underbrace{\boldsymbol{\beta}}_{\text{Uncertainty penalty}} + \log 2. \tag{3}$$

The scale head is trained using a batch size of 8 and a learning rate of 5×10^{-4} over $1,000$ training iterations.

3.2. Details of Reward Head

Our VLM-guided instant reflector integrates a pretrained vision-language model (LLaVA [4]) with a lightweight reward head that predicts continuous reward signals, see Figure 2 for an overview. The VLM backbone is kept frozen to preserve its semantic capabilities, and only the reward head is trained. Given a video sequence $\{f_1, \ldots, f_N\}$ generated by the world simulator, we uniformly sample 32 frames as visual input. The language prompt is formatted as: *"Watch the video and determine whether it completes the task:* $\{\mathbf{g}\}$ *— answer only 'Yes' or 'No'."* The VLM processes this input and extracts a pooled embedding, which is projected by the reward head to a scalar. A sigmoid activation yields a continuous reward $R \in [0, 1]$, interpreted as the task completion probability. The reward head is trained with binary cross-entropy loss, using a batch size of 8, learning rate 1×10^{-4}, Adam optimizer, and 50 epochs, with input frames center-cropped to 384×384 resolution.

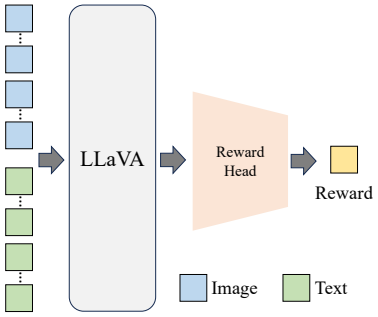


Figure 2. **Network architecture of instant reflector.**

Algorithm 1 RehearseVLA Training Algorithm

Input: Pretrained VLA policy π_θ, scale head β_θ, VLM-based reward function $R(\mathbf{o}_{1:t}, \mathbf{g})$, context dataset $\mathcal{D}_{\text{context}}$

for training iteration $= 1$ to M **do**
  Set behavior policy: $\pi_\phi \leftarrow \pi_\theta, \quad \beta_\phi \leftarrow \beta_\theta$ ▷ Fix old policy and scale head
  Initialize rollout buffer $\mathcal{D}_{\text{rollout}} \leftarrow \emptyset$
  while $|\mathcal{D}_{\text{rollout}}| < B$ **do** ▷ Rollout Collection
    Sample context $\mathbf{c} = (\mathbf{g}, \mathbf{o}_1, \mathbf{s}_1) \sim \mathcal{D}_{\text{context}}$
    for $n = 1$ to N **do** ▷ Generate N rollouts per context
      Initialize trajectory $\tau_n \leftarrow (\mathbf{o}_1, \mathbf{s}_1)$
      for $t = 1$ to T **do**
        Predict base action: $\boldsymbol{\mu}_t \leftarrow \pi_\phi(\mathbf{o}_{1:t}, \mathbf{s}_{1:t}, \mathbf{g})$
        Predict log-scale: $\boldsymbol{\beta}_t \leftarrow \beta_\phi(\mathbf{o}_{1:t}, \mathbf{s}_{1:t}, \mathbf{g})$
        Sample action: $\mathbf{a}_t \sim \text{Laplace}(\boldsymbol{\mu}_t, \exp(\boldsymbol{\beta}_t))$
        Compute next proprioceptive state: $\mathbf{s}_{t+1} \leftarrow \text{FK}(\mathbf{s}_t, \mathbf{a}_t)$
        Predict next observation: $\mathbf{o}_{t+1} \leftarrow \text{WorldSim}(\mathbf{o}_t, \mathbf{s}_{t+1})$
        Append $(\mathbf{a}_t, \mathbf{o}_{t+1}, \mathbf{s}_{t+1})$ to τ_n
        if $R(\mathbf{o}_{1:t+1}, \mathbf{g}) > \eta$ **then** ▷ Termination check ($\eta = 0.5$)
          $t_{\text{end}} \leftarrow t + 1$; **break**
        end if
      end for
      Set trajectory reward: $R_n \leftarrow R(\mathbf{o}_{1:t_{\text{end}}}, \mathbf{g})$
      Store log-probabilities $\log p_\phi(\mathbf{a}_{1:t_{\text{end}}} \mid \cdot)$ for importance weighting
    end for
    Compute RLOO baselines: $b_n \leftarrow \frac{1}{N-1} \sum_{j \neq n} R_j$ for all n
    Compute advantages: $A_n \leftarrow R_n - b_n$
    Add $\{(\tau_n, A_n, \log p_\phi(\cdot))\}_{n=1}^{N}$ to $\mathcal{D}_{\text{rollout}}$
  end while
  for optimization step $= 1$ to K **do**
    Sample batch from $\mathcal{D}_{\text{rollout}}$
    Compute current log-probabilities $\log p_\theta(\mathbf{a} \mid \cdot)$
    Compute importance ratios: $r_t \leftarrow \exp(\log p_\theta - \log p_\phi)$
    Update π_θ and β_θ by minimizing PPO loss
  end for
end for

Table 2. **Performance of the instant reflector under different η.** The selected value $\eta = 0.5$ achieves a balanced trade-off across accuracy, precision, recall, and F1-score.

η	Accuracy	Precision	Recall	F1-score
0.2	0.825	0.667	1.0	0.8
0.4	0.875	0.764	0.928	0.838
0.5 (Ours)	0.925	0.867	0.928	0.896
0.6	0.875	0.846	0.785	0.815
0.8	0.85	0.9	0.642	0.75

4. Analysis of Instant Reflector

To investigate the sensitivity of model performance to the hyperparameter η, we conduct a series of experiments by varying η over a predefined range (e.g., 0.2, 0.4, 0.5, 0.6, 0.8). As shown in Table 2, we report four standard evaluation metrics: accuracy, precision, recall and F1-score. The results indicate that $\eta = 0.5$ yields consistently strong performance across all metrics.

5. Analysis of World Simulator

5.1. Data Analysis and Distribution

We provide a statistical analysis of the training data for the world simulator and instant reflector in Figure 3, including: (a) length distributions for successful vs. failed trajectories, (b) cumulative distribution functions by outcome, and (c) task outcome proportions. The bimodal distribution in successful trajectories motivated our dynamic termination mechanism, while the long-tailed length distribution informed our curriculum sampling strategy.

5.2. Ablation Studies

We evaluate the impact of training data, our proposed geometry-aware VGGT feature injection, and alternative representations (such as SAM and DINO) on the performance of the world simulator. Quantitative results are presented in Table 4, where "w/o extra" denotes the model trained without additional data, and "w/o VGGT" refers to the variant without our geometry-aware feature injection strategy. We evaluate using standard metrics: FID [1], FVD [6], PSNR [2], SSIM [7], and LPIPS [8]. As shown in Table 4, both the expanded training data and the VGGT latent features significantly contribute to building a more robust and physically consistent world model. Additional, we perturb the world model's outputs with: (i) Gaussian noise (mean = 0, variance = 0.1), and (ii) color perturbation, randomly adjusting brightness ($\pm 20\%$), contrast ($\pm 20\%$), saturation ($\pm 20\%$), and hue (± 0.1). As shown in Table 3, thanks to the powerful pretrained VLM and our data augmentation, these perturbations only incur minor performance drop, validating our robustness to imperfect early-stage predictions.

Task	Gaussian noise	Color perturbation	Original (Ours)
LIBERO-Spatial	85.4	87.0	87.6

Table 3. **Impact of world model perturbations on final results**

5.3. More Results

Figures 4 and 5 show additional trajectories generated by the world simulator, demonstrating its ability to synthesize both successful and failed task executions.

References

[1] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 3

[2] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13): 800–801, 2008. 3

[3] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *RSS*, 2025. 1

[4] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1

[5] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 1

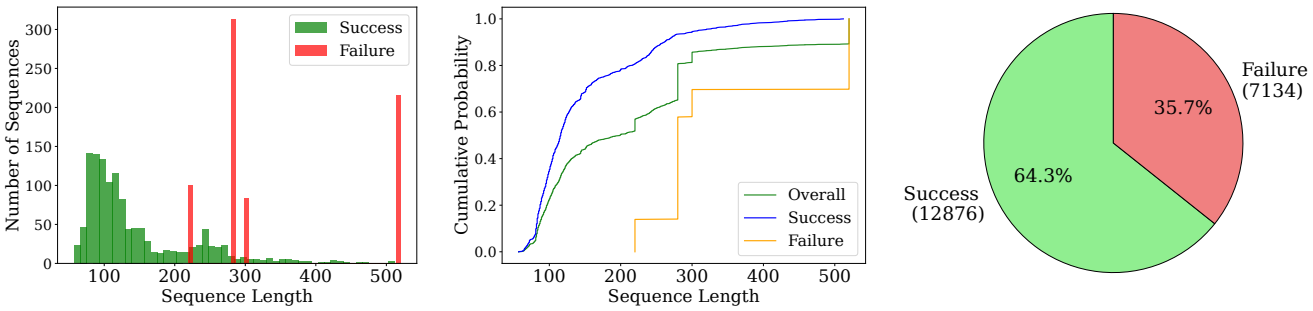
[6] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. 3

[7] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 3

[8] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3

Table 4. **Quantitative comparison of world models.**

Method	FID↓	FVD↓	PSNR↑	SSIM↑	LPIPS↓
w/o extra	100.9978	116.7792	20.1600	0.7450	0.1385
w/o VGGT	70.9232	114.5264	23.1841	0.8219	0.0806
Ours with DINO	41.7398	74.9209	23.4069	0.8511	0.0602
Ours with SAM	40.6276	74.0298	23.6022	0.8532	0.0599
Ours (CLIP+VGGT)	**39.1941**	**73.5313**	**23.8343**	**0.8579**	**0.0562**



(a) Length Distribution (b) Cumulative Distribution Func. (c) Task Outcome Proportion

Figure 3. **Training data analysis and distribution.**

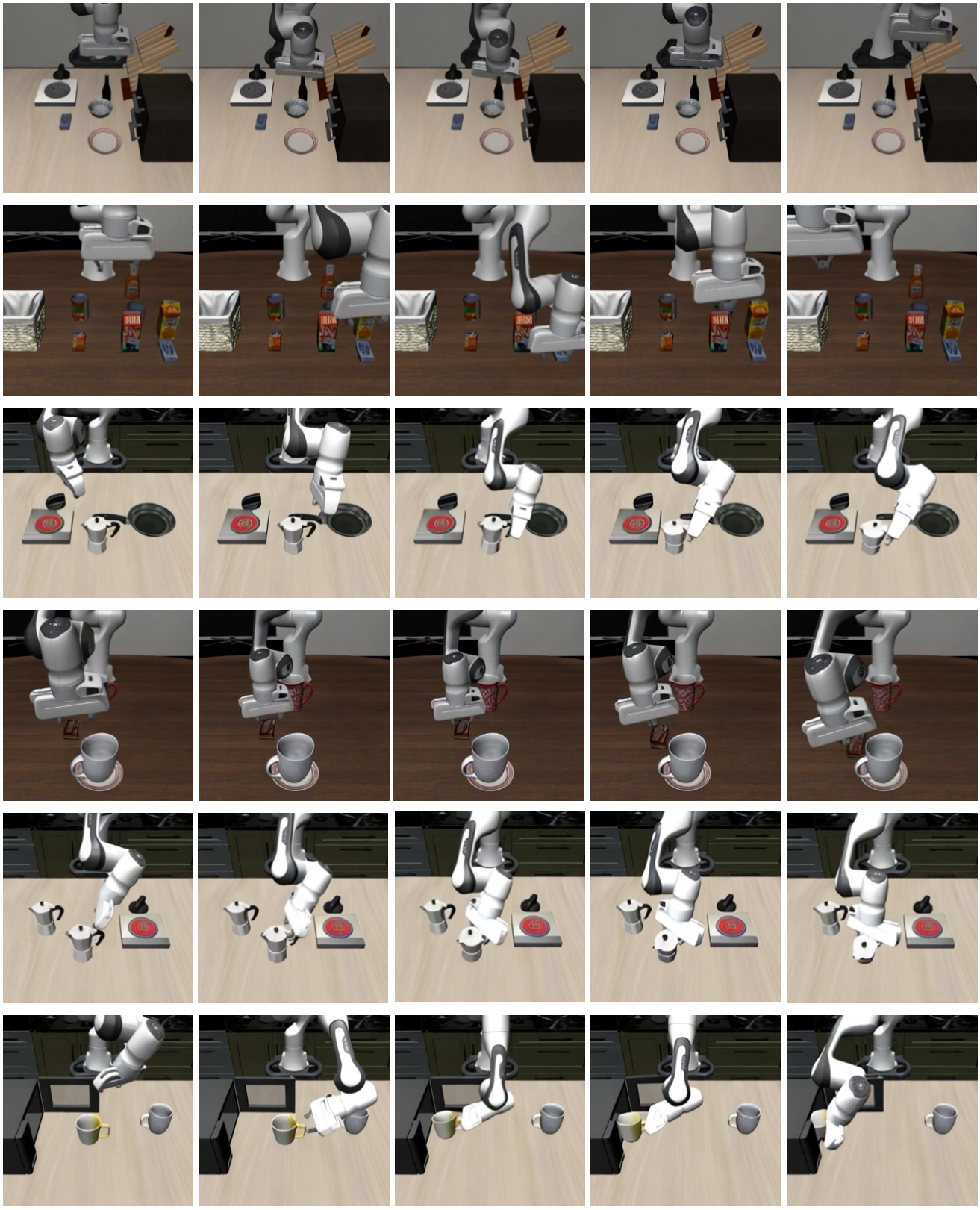


Figure 4. **Failure trajectories synthesized by the world simulator.**

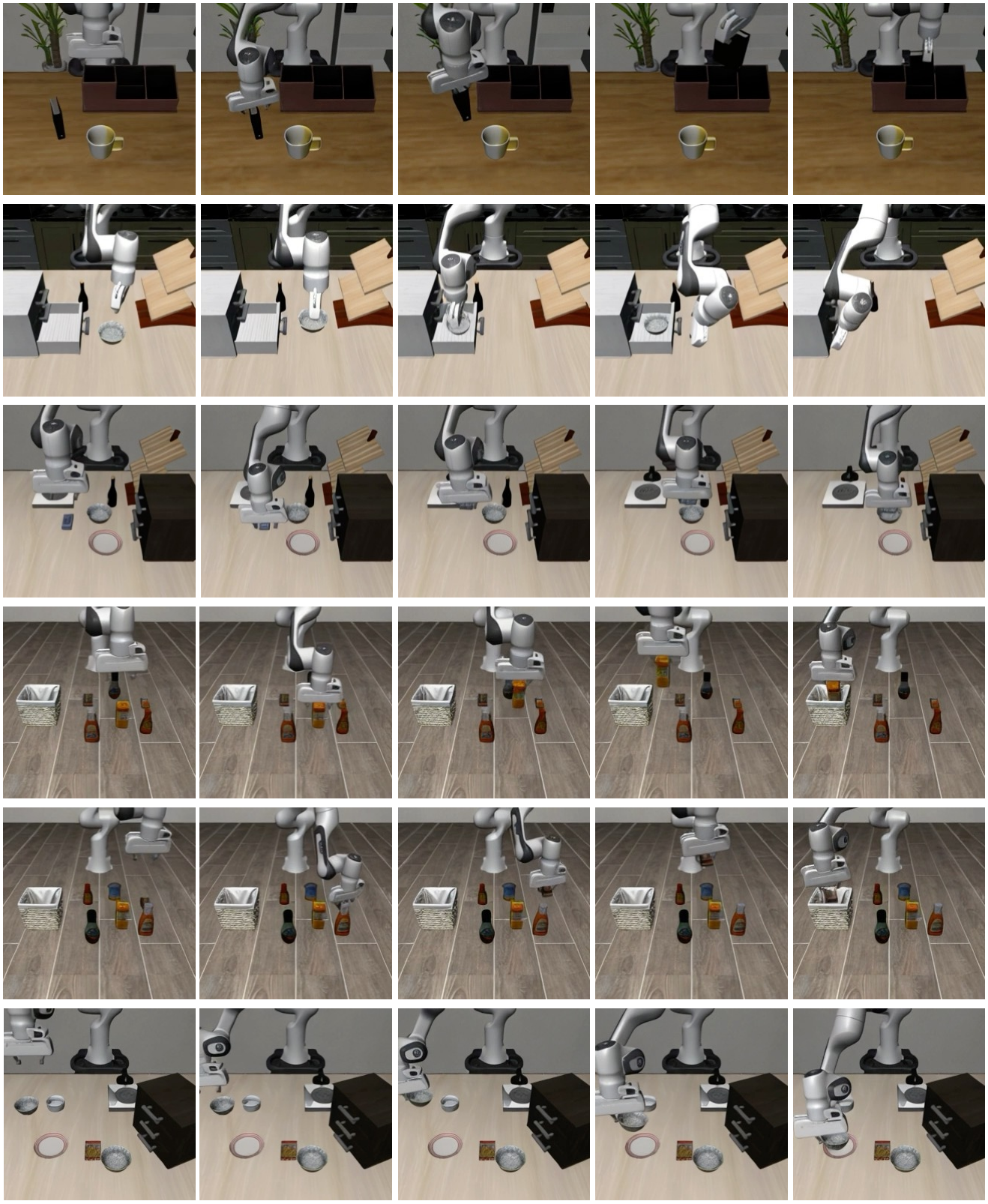


Figure 5. **Success trajectories synthesized by the world simulator.**