ENERVERSE-AC: Envisioning Embodied Environments with Action Condition

Yuxin Jiang[1,*], **Shengcong Chen**[1*], **Siyuan Huang**[2*], **Liliang Chen**[1†], **Pengfei Zhou**[1], **Yue Liao**[3]
Xindong He[1], **Chiming Liu**[1], **Hongsheng Li**[4], **Maoqing Yao**[1 ‡], **Guanghui Ren**[1 ‡]
[1]AgiBot [2]SJTU [3]LV-NUS Lab [4]CUHK MMLab
Project Page: `https://annaj2178.github.io/EnerverseAC.github.io`
Email: yaomaoqing@agibot.com, renguanghui@agibot.com [*]

Abstract

Robotic imitation learning has advanced from solving static tasks to addressing dynamic interaction scenarios, but testing and evaluation remain costly and challenging due to the need for real-time interaction with dynamic environments. We propose ENERVERSE-AC (abbr. EVAC), an action-conditional world model that generates future visual observations conditioned on an agent's predicted actions, enabling realistic and controllable robotic inference. Building on prior architectures, EVAC introduces a multi-level action-conditioning mechanism and ray map encoding for dynamic multi-view image generation while expanding training data with diverse failure trajectories to improve generalization. As both a data engine and evaluator, EVAC augments human-collected trajectories into diverse datasets and generates realistic, action-conditioned video observations for policy testing, reducing the evaluation costs post-training. This approach significantly reduces costs while maintaining high-fidelity robotic manipulation evaluation. Extensive experiments validate the effectiveness of our method. Code, checkpoints, and datasets will be released. For further visualization results, we strongly recommend visiting the Project Page.

1 Introduction

The development of robotic imitation learning has significantly advanced robotic manipulation, transitioning the field from solving isolated tasks in static environments to addressing complex and diverse interaction scenarios. Unlike traditional AI domains such as computer vision (CV) or natural language processing (NLP), where model performance can be evaluated using non-interactive and static datasets, robotic manipulation inherently requires real-time interaction between agents and dynamic environments during testing and evaluation. As task diversity grows, assessing policy performance often necessitates direct deployment on physical robots or the creation of large-scale 3D simulation environments, both of which are costly, labor-intensive, and challenging to scale.

Building low-cost, scalable testing and inference environments for robotic manipulation has thus become a critical challenge in robotic imitation learning. Recently, the concept of using video generation models as world simulators has emerged as a promising direction. These models enable agents to observe and interact with dynamic worlds through learned visual dynamics, circumventing the need for explicit physical simulation. While this approach introduces a new avenue for constructing robotic inference pipelines, existing world modeling techniques primarily focus on generating videos from language instructions and predicting actions based on the generated videos. However, these methods fall short of creating true world simulators, which should simulate environment dynamics in response to the agent's actions, enabling realistic and controllable testing.

[*]∗ indicates equal contribution. † indicates project leader. ‡ indicates corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: EWM.

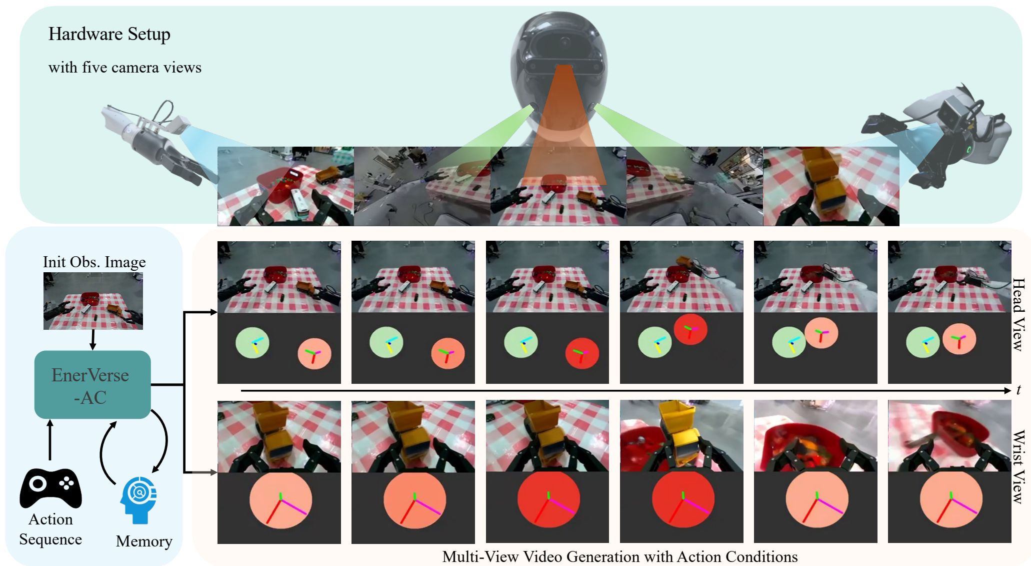


Figure 1: Overview of the EVAC framework. Given initial observation images and an action sequence, EVAC generates multi-view videos conditioned on the provided actions. By incorporating a memory mechanism, EVAC supports the generation of long-term video sequences. The framework handles both static head camera views and dynamic wrist camera views to provide a comprehensive representation of the robotic environment.

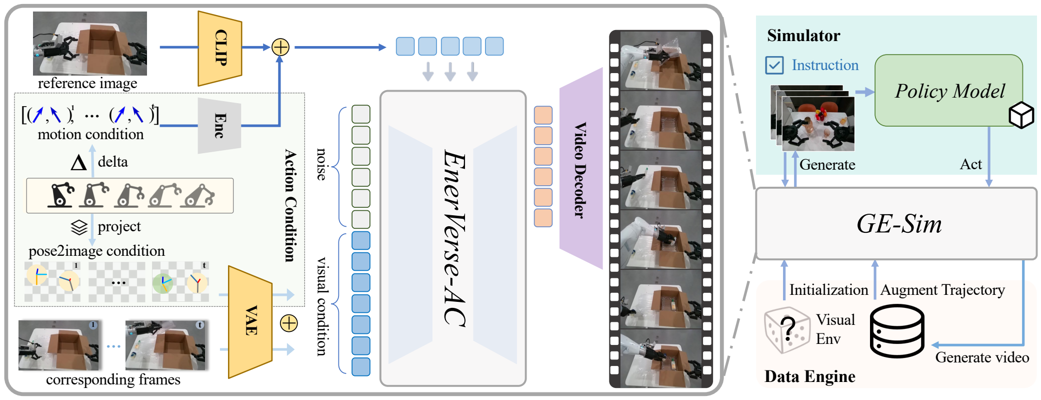
To bridge this gap, we propose EVAC, an action-conditional world model that generates future visual observations directly conditioned on the agent's predicted actions. Built upon prior embodied world model architectures like **EnerVerse** [11], ENERVERSE-AC incorporates additional **A**ction-**C**onditioning information to enable more realistic and controllable robotic inference. To achieve this, we designed a multi-level action condition injection mechanism, which uses end-effector projection action maps and delta action encodings. Furthermore, to support the generation of multi-view images, crucial for embodied tasks, we introduce spatial cross-attention modules and ray direction map encoding to process multi-view features. To reflect the movement of camera, we encode the camera's motion using ray map embeddings.

Beyond architectural innovations, the EVAC world model is designed to handle both successful and failure scenarios. In addition to leveraging the Agibot-World dataset [5], we curated a diverse dataset of failure trajectories, significantly expanding the training data's coverage. This enhancement improves the model's ability to generalize across diverse scenarios, ensuring its applicability to real-world robotics tasks.

The proposed EVAC world model serves as both a data engine for policy learning and an evaluator for trained policy models, addressing key challenges in robotic manipulation. As a data engine, EVAC augments limited human-collected trajectories into diverse datasets by segmenting actions (e.g., fetching, grasping, homing), applying spatial augmentations, and generating new video sequences, thereby enhancing policy robustness and generalization. As an evaluator, it eliminates the need for complex simulation assets by generating realistic, action-conditioned video observations for iterative policy testing, which can be reviewed by human evaluators or automated systems like Video-MLLMs. This approach significantly reduces reliance on real robot hardware during development, saving costs and time, while maintaining high evaluation fidelity correlated with real-world performance.

2 Related Work

Video Generation Model as World Model. While prior research on generative models has shown promise, [3, 4] highlights video generation as an innovative approach to constructing world models. Similarly, [32] aims to develop a universal world model built upon the generative model but focuses on generating only the next-step frames rather than continuous video sequences. Video generation remains a challenging task with applications across diverse domains. Recent advancements in



(a) Action-conditioned EVAC Framework **(b) Simulator and Data Engine**

Figure 2: Overview of the EVAC Framework. (a) The framework begins with a reference image, whose feature vector serves as the reference style guidance. The original robotic actions are processed to compute the delta action vector and this temporal information is concatenated with the reference style guidance and injected into the diffusion model via a cross-attention mechanism. Additionally, the action information is projected into action maps, whose feature maps are concatenated with feature maps from both memory and visual observations before being fed into the diffusion network. The diffusion model generates video frames with denoising process, followed by a video decoder to produce the final output. For simplicity, we only demonstrate the single-view case here. (b) With this action-following ability, EVAC could be used as the policy simulator and data engine.

diffusion models [9] and latent diffusion models [24] have demonstrated progress in generating high-quality images with reduced computational complexity. Furthermore, text-guided and pose-guided video generation methods [18, 10] have expanded the applicability of video synthesis technologies.

In robotics, works like [34, 11] focus on generating future frames from textual and visual inputs. However, limited attention has been given to video generation conditioned on robotic actions. Gesture-conditioned approaches [27] provide valuable insights but have yet to be tested in robotics, where environments and object interactions are significantly more complex. Advancements in action-conditioned video generation are essential to address these challenges.

Physical Simulators for Robotics. Physical simulators are widely applied in robotics learning tasks. MuJoCo [26] has been used for locomotion and manipulation studies, while PyBullet [7] supports real-time control and sim-to-real experiments. Similarly, Isaac Gym [19] facilitates reinforcement learning in continuous control tasks with large-scale parallel environments. Several studies [20] utilize physical simulators to train policies for solving dexterous manipulation tasks. Despite their utility, physical simulators face notable limitations. The sim-to-real gap often results in overfitting to synthetic environments, reducing real-world performance. Moreover, creating digital assets—including robot embodiments, target objects, and task scenes—remains labor-intensive and requires expert-level effort, further hindering scalability.

Robotics Imitation Learning. Recent advancements in robotics imitation learning focus on developing generalist models capable of efficiently handling diverse tasks across multiple embodiments using extensive multimodal datasets. Models such as RT-1 [2], Gato [21], Octo [25], and OpenVLA [14] integrate pretrained visual and language models with specialized policy heads, enabling remarkable task generalization. Building on this, [6] introduces a dual-brain system, while [22] employs layer-wise information with flow matching techniques for action prediction. Additionally, [5] transitions from direct action prediction to latent action representations, ensuring more effective generalization. However, these approaches rely heavily on large-scale action datasets for training. While some works, such as [12], attempt to reduce data requirements by increasing information density, they still depend significantly on human data collection, underscoring the need for further innovations in data-efficient learning techniques.

3 Method

In EVAC, we adopt a UNet-based video generation model as our baseline, following [31, 11]. Beyond this, we propose an action-conditioned framework, as illustrated in Figure 2. Given an RGB video set $O \in \mathbb{R}^{V \times (H+K) \times 3 \times h \times w}$, where V denotes the number of views, H represents the number of observed history frames, K is the number of intended predicted frames, and h, w are the frame height and width, our method is designed to predict future frames based on observed past frames and robotic actions. First, we pass the video set through an encoder ε to obtain the latent representation $z \in \mathbb{R}^{V \times H \times C \times h \times w}$, where C is the latent dimensionality. Using a latent diffusion model, we aim to predict $z_t = p_\theta(z_{t-1}, c, t)$, where c is the condition signal and t is the denoising timestep. In this work, the conditioning signal originates from the robotic action trajectory $A \in \mathbb{R}^{(H+K) \times d}$, where $d = 7$ represents the end-effector pose with $[x, y, z, roll, pitch, yaw, openness]$ and $d = 14$ in bi-arm case.

To inject the action condition, we use both spatial-aware pose information injection and delta action attention module. Furthermore, we extend traditional 2D video generation to 3D video generation, represented by multi-view frames, to better meet the requirements of robotic manipulation tasks.

3.1 Mutli-Level Action Condition Injection

Spatial-Aware Pose Injection. [30], [29], [28], [10] have proposed different ways on controlling video generation by injecting pose information. One common way to align the image with the fine-grained pose trajectory is to use a pixel-alignment method to inject the pose signal. In the field of robotics, 6D pose has been tested as an effective representation of action space. Therefore, to ensure precise visual alignment with the conditioned image, we have developed methodologies to effectively depict the 6D end effector pose of the end effector. First, we convert the end-effector position at timestamp i in world coordinates to the corresponding pixel coordinates using the calibrated camera parameters. Furthermore, to visually represent the roll, pitch, and yaw angles in 2D image space, we employ visual prompting techniques inspired by [15, 16]. This approach utilizes unit vectors along each directional axis, providing an intuitive representation of the end-effector's orientation.

To illustrate the gripper action at each state, we use a unit circle to encode the action magnitude, where lighter shades correspond to open gripper and darker shades indicate closed gripper. To differentiate between the left and right hand, we employ distinct color schemes for visualizing 6D poses and gripper actions. The 6D pose visualization is rendered on a black background to enhance clarity, as shown in Figure 3. After constructing the action map using the aforementioned visual prompting techniques, we process it with the CLIP [23] vision encoder. The resulting feature maps are concatenated with the feature maps from RGB images along the channel dimension.

Delta Action Attention Module. Furthermore, we designed a Delta Action Attention module which calculates the delta motion between consecutive frames to approximate changes in the end-effector's position and orientation. These delta motions are encoded into a fixed number of latent representations by a linear projector and then via cross-attention [1], [13]. The fixed-length latent representation token is then fused with the reference-image features (e.g. the first input frame features) and injected into the Unet stage through a cross-attention mechanism. By incorporating temporal changes, such as speed and acceleration, the module enhances the model's physical understanding of motion dynamics, enabling it to produce more realistic and diverse video outputs.

3.2 Multi-View Condition Injection

In embodied robotics, cross-view information, particularly visual inputs from wrist cameras, is essential for accurate trajectory prediction. To address this, we extend EVAC world model to support multi-view video generation. Following EnerVerse [11], multi-view features are fed where spatial cross-attention modules enable interaction between views. A ray direction map encoding camera parameters is also concatenated into the input features to provide spatial context. Unlike EnerVerse, which processes only static camera views, EVAC incorporates dynamic wrist camera views that move together with the robotic arms. This creates a challenge: when projecting end-effector (EEF) poses onto wrist camera images using Section 3.1 methods, the projection circle remains static, failing to convey the hand's movement, as shown in Figure 3. Inspired by techniques in [8, 11], we encode camera motion using the origins o_r and directions d_r of ray maps $r = (o_r, d_r)$. Specifically, for each camera, we compute the ray maps relative to its poses at all times. Since the wrist cameras move with the arms, the ray maps of the wrist cameras can implicitly encode the motion information of EEF poses. Therefore, the ray maps are concatenated with the trajectory maps to provide enriched trajectory information, improving cross-view consistency.

Figure 3: Visualizing EEF Projections and the Ray Maps. The bottom row illustrates wrist camera views, where projections appear nearly identical. Then, ray maps provide additional spatial context to represent movements. The value of the ray maps is visualized with the RGB value.

3.3 Applications

Data Engine for Policy Learning. Beyond evaluation, EVAC can serve as a data engine for robotic policy learning by generating diverse training trajectories from limited demonstrations. Inspired by [33], we develop a systematic approach to augment manipulation datasets through controlled trajectory generation. We demonstrate our approach using primitive pick-and-place tasks as an illustrative example. Given M human-collected trajectories, we first segment each trajectory into three distinct phases—fetching, grasping, and homing—by analyzing gripper openness changes to identify contact timestamps t_b (beginning) and t_e (ending).

For trajectory augmentation, we focus on the fetching phase and extract the visual observation O_{t_b} along with the corresponding action sequence $[a_{t_{b-N}}, \ldots, a_{t_b}]$, where $N = 90$ frames (3 seconds before grasping). Our augmentation process follows three key steps: **(1) Spatial Augmentation:** We keep the final action a_{t_b} fixed while spatially perturbing the initial action $a_{t_{b-N}}$ within a predefined range to generate $a'_{t_{b-N}}$, creating diverse starting conditions for the fetching motion. **(2) Trajectory Generation:** We interpolate between $a'_{t_{b-N}}$ and a_{t_b} to create smooth action sequences, then feed the reversed sequence $[a_{t_b}, \ldots, a'_{t_{b-N}}]$ along with O_{t_b} into EVAC to generate corresponding video frames. **(3) Dataset Reconstruction:** We reorder the generated frames to create properly sequenced training data, ensuring temporal consistency for policy learning.

This process transforms the original M trajectories into a significantly more diverse dataset with varied approach trajectories while maintaining task-relevant endpoints. Figure 4 illustrates this augmentation process, showing how different initial positions (indicated by red arrows) lead to diverse fetching motions that converge to the same grasping configuration.

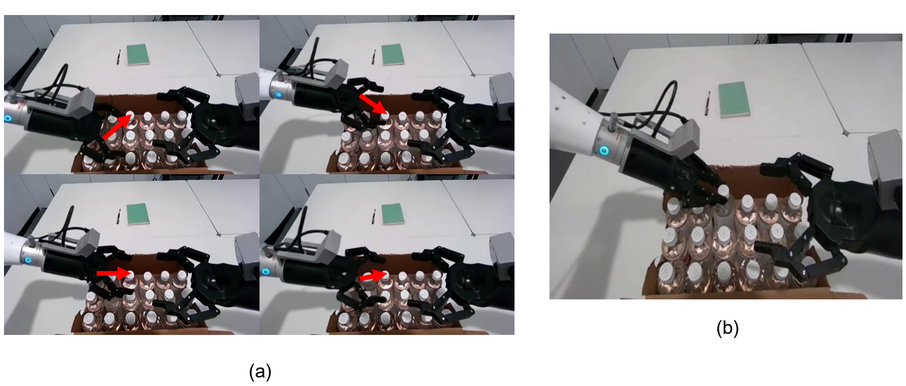


Figure 4: Data augmentation process visualization. **Left**: Spatially augmented initial actions $a'_{t_{b-N}}$ with four example starting frames. Red arrows indicate diverse approach directions toward the target grasping position. **Right**: Fixed final action a_{t_b} representing the consistent grasping configuration. Intermediate frames are generated through linear interpolation and EVAC world model prediction.

Evaluator for Policy Model. Another application of EVAC is to serve as a physical simulator for evaluating trained policy models. Given an initial visual observation O_t and corresponding instructions, the policy model generates action chunks. We then feed these action chunks, along with O_t, into the EVAC world model to generate new observations. This process is repeated iteratively until the action norm generated by the policy model falls below a predefined threshold, at which point evaluation terminates automatically. This threshold mechanism simulates policy 'hesitation' and maintains evaluation uniformity between EVAC and real-world robot testing. Subsequently, multiple human evaluators assess task success by watching the EVAC-generated videos.

This evaluation approach offers two key advantages. First, it eliminates the need to create complex simulation assets. Second, the video replay can be sped up to save time, or it can potentially be integrated with video-based Video-MLLMs, reducing the need for human evaluation efforts. By leveraging this process, the EVAC world model can largely replace the use of real robot hardware during the initial development stage, significantly reducing deployment efforts. Our experiments reveal a high correlation between evaluation results obtained through EVAC and those observed in real-world scenarios.

4 Experiments

4.1 Experiment Details

Dataset The training data for EVAC is primarily derived from the AgiBot World dataset [5], which contains over 210 tasks and 1 million trajectories. To ensure comprehensive coverage of action trajectories, including both successful and failed cases that are critical for enabling EVAC to function as a generalized simulator, we collaborated with the AgiBot-Data team to obtain full access to the raw data. We developed an automated data collection pipeline to capture real-world failure cases during teleoperation and real-robot inference. The resulting dataset includes approximately 100,000 failure trajectories (10% of the total 1 million trajectories) that were automatically logged with predefined triggers (teleop aborts, anomalous contact, etc). These failure cases reflect authentic real-robot anomalies, thereby minimizing distribution mismatch between training data and real-world deployment scenarios.

Implementation Details Our model is built on UNet-based Video Diffusion Models (VDM) [31]. During training, the CLIP visual encoder and VAE encoder are frozen, while other components, including the UNet, resampler, and linear layers, are fine-tuned. The model is trained with a batch size of 16. For the single-view version, training requires approximately 32 A100 GPUs for 2 days, whereas the multi-view version takes about 32 A100 GPUs for 8 days. We experimentally determined that setting the memory size to 4 and the chunk size to 16 achieves a balance between generation quality and resource cost. The memory consists of 4 historical frames, each derived from the results of the previous chunk generation. For the robotic policy model, we utilize the official single-view version of GO-1 [5]. For inference, EVAC requires 8s per 16-frame chunk (DDIM 27 steps) on an RTX 4090. The concatenated conditions include repeated latent features of the condition frame, action maps, ray maps, and a dropout mask indicating whether the condition is dropped. This dropout strategy is designed to improve the model's robustness. The hyperparameters of the model architecture and training setup are provided in Appendix A.

4.2 Controllable Manipulation Video Generation

As shown in Figure 5, EVAC excels at synthesizing realistic videos of complex robot-object interactions, even in challenging scenarios. A key strength of EVAC lies in its ability to maintain high visual fidelity while accurately following input action trajectories, ensuring reliability for building credible evaluation systems.

The model's chunk-wise autoregressive diffusion architecture and sparse memory mechanism, inspired by [11], enable it to sustain visual stability and scene consistency during continuous chunk-wise inference. Experimental results show that the generated videos remain sharp and reliable for up to 30 consecutive chunks in single-view scenarios and 10 chunks in multi-view settings. However, artifacts and blurring begin to emerge in longer sequences, highlighting a tradeoff between sequence length and visual quality. Figure 6 further illustrates EVAC's ability to preserve scene integrity across multiple chunks during a manipulation task. The snapshots showcase environment consistency over time, demonstrating EVAC's robust performance in maintaining visual coherence during chunk-wise autoregressive inference.

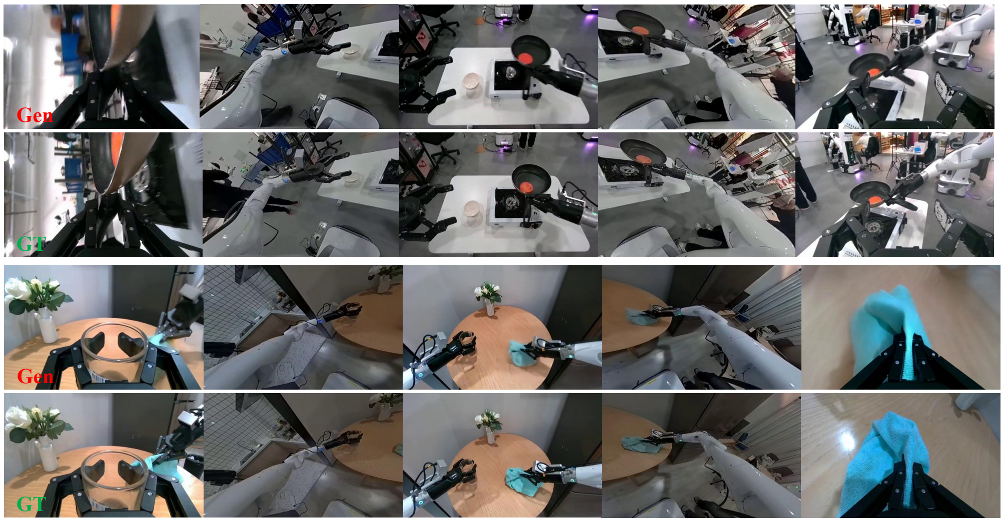


Figure 5: Qualitative results for multi-view video generation. The figure shows EVAC's ability to generate consistent multi-view videos conditioned on robotic actions, with GT (ground truth) and Gen (generated) sequences displayed for comparison.

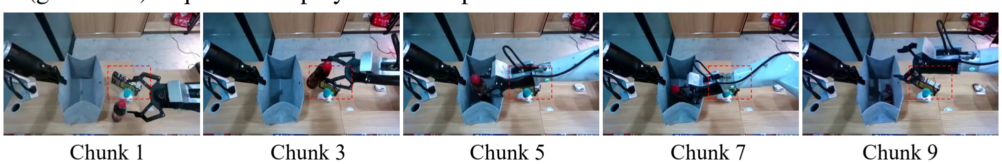


Figure 6: Environment consistency during chunk-wise inference. Snapshots from EVAC at various inference stages (Chunks 1, 3, 5, 7, and 9) demonstrate robust performance in maintaining visual fidelity and scene coherence over time.

4.3 EVAC as Policy Evaluator

This section evaluates EVAC's effectiveness as a policy evaluator by assessing the consistency between EVAC-generated simulations and real-world robot performance. Our evaluation addresses a critical challenge in robot learning: the need for reliable, cost-effective policy assessment without extensive real-world testing.

Experimental Setup. We select four diverse manipulation tasks (Figure 12 in Appendix B) and train the single-view GO-1 policy [5] without the latent planner module. Each task is evaluated under slightly randomized initial conditions to ensure generalization, with 40 trials per task in both real-world and EVAC environments. Real-world evaluations are conducted first, and the initial frame recordings serve as image conditions for corresponding EVAC evaluations. Success is determined by three independent evaluators who assess whether the robot successfully retrieves the target item.

Cross-Task Performance Consistency. Figure 7 (left) demonstrates that while absolute success rates show minor differences between EVAC and real-world evaluations, the relative performance trends across tasks remain highly consistent. This consistency validates EVAC's reliability for cross-task policy performance analysis and its ability to accurately replicate real-world dynamics rankings.

Training Dynamics Tracking. Robot policy learning often exhibits performance fluctuations across training steps. To evaluate EVAC's ability to capture these dynamics, we assess the same policy at different training checkpoints using the "Take a Bottle" task. Figure 7 (right) shows that both EVAC and real-world evaluations capture identical performance trends, with success rates improving consistently as training progresses. This result confirms EVAC's capability to accurately mirror real-world performance variations during policy development.

Failure Mode Analysis and Cross-Environment Validation. Expert evaluation reveals that failure modes in EVAC (e.g., empty grasps, unintended collisions) align with actual robot failures in 78.6% of cases. To further validate EVAC's generalization capabilities, we conduct cross-environment comparisons using the LIBERO simulator. Figure 8 presents side-by-side comparisons between

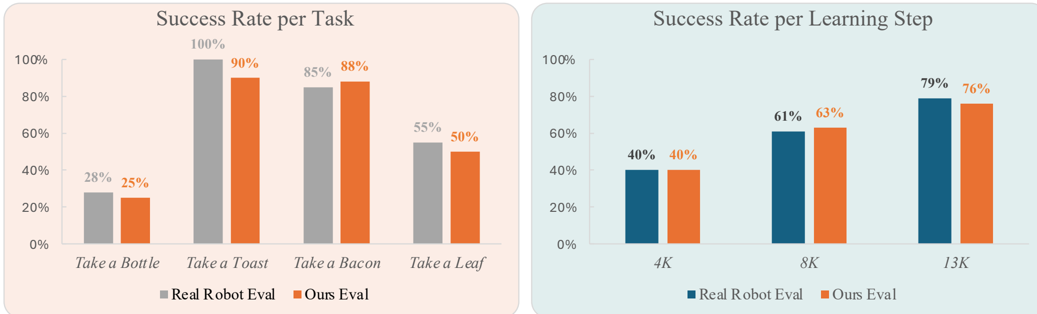


Figure 7: Policy evaluation consistency across tasks and training progression. **Left**: Cross-task performance comparison showing consistent relative rankings between EVAC and real-world evaluations across diverse manipulation tasks. **Right**: Training dynamics comparison showing aligned performance trends between EVAC and real-world testing across different training checkpoints.

LIBERO simulator outputs and EVAC-generated sequences for identical action inputs. Notably, both environments consistently identify the same failure modes—particularly collision events highlighted in red rectangles—demonstrating EVAC's reliability in detecting critical policy failures across different visual domains.

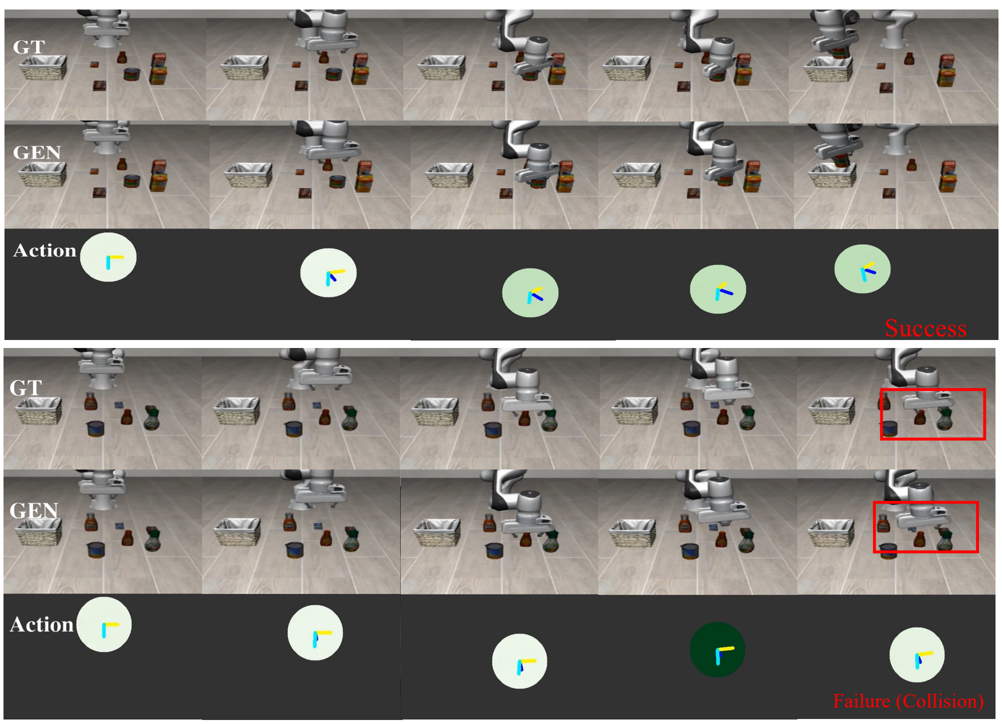


Figure 8: Cross-simulator validation comparing LIBERO simulator and EVAC with identical action trajectory inputs. Collision-induced failure cases are consistently captured by both systems (highlighted with red rectangles), validating EVAC's reliability for policy evaluation across different simulation environments.

Unlike conventional simulators that suffer from domain gaps in lighting and texture rendering, EVAC operates directly on real-world image inputs, delivering reproducible evaluations under consistent visual conditions while eliminating expensive simulation environment creation costs (Figure 9).

4.4 EVAC as Data Engine

This section demonstrates EVAC's capability to generate novel action trajectories for policy training data augmentation, leading to improved task performance across different environments and policy

Figure 9: Evaluation environment comparison: EVAC vs. conventional simulators for the same manipulation task, highlighting visual fidelity and setup complexity differences.

architectures. To comprehensively evaluate this capability, we conduct experiments in both real-world and simulated environments using different VLA models.

Experimental Setup. We design a systematic evaluation comparing baseline training (using only ground truth demonstrations) against augmented training (incorporating EVAC-generated synthetic trajectories). Two experimental settings are employed: (1) **Real-world evaluation** using the GO-1 policy [5] on a challenging bottle extraction task that requires extracting a tightly packed water bottle from a paper box and placing it on a table; and (2) **Simulated evaluation** using OpenVLA [14] on the LIBERO Spatial [17] to assess performance across diverse manipulation tasks.

Results and Analysis. As shown in Table 1, data augmentation with EVAC consistently improves performance across both settings. In the real-world setting, incorporating 30% synthetic trajectories alongside 20 expert demonstrations improves the success rate from 0.28 to 0.36 (28.6% relative improvement). In the simulated environment, progressive augmentation with 5, 10, and 20 synthetic episodes yields steady performance gains, with success rates improving from 58.2% to 66.0% (13.4% relative improvement). These results demonstrate EVAC's effectiveness in enhancing policy learning with limited expert data across different environments, tasks, and model architectures, validating its utility as a versatile data engine for robotic manipulation.

Table 1: Effectiveness of Data Augmentation on Policy Training Performance

Environment	**VLA Model**	**Data Configuration**	**SR**
Real World	GO-1	Baseline (20 GT only)	28.2%
		+ 6 Synthetic	36.0%
Simulator	OpenVLA	Baseline (20 GT only)	58.2%
		+ 5 Synthetic	61.6%
		+ 10 Synthetic	64.2%
		+ 20 Synthetic	66.0%

4.5 Further Analysis

Failure Data Matters. As discussed in Section 4.1, we deliberately collected failure trajectories to expand the action coverage in the training data. To evaluate the effectiveness of this failure data, we trained two models: one with failure trajectories included and the other without. As illustrated in Figure 10, we tested the models using a scenario where the robotic arm was pretending to grasp a bottle of water that was not actually present.

Without failure data, the model tended to overfit to successful examples, leading it to "hallucinate" that the bottle had been successfully grasped despite the absence of physical interaction. In contrast, with the inclusion of failure data, EVAC was able to accurately recognize and distinguish the failed grasp attempt, demonstrating its robustness against overfitting and its ability to handle edge cases effectively.

The effectiveness of Delta Action Attention Module To ensure the generated videos from EVAC accurately follow action trajectories, we employ a Multi-level Action Condition Injection strategy. To assess the impact of the Delta Action Attention Module, we conducted ablation studies, with results shown in Figure 11. The task involved intricate pan manipulation dynamics, including rapidly shaking the pan, slowly shaking the pan, upward tossing, and upward shaking.

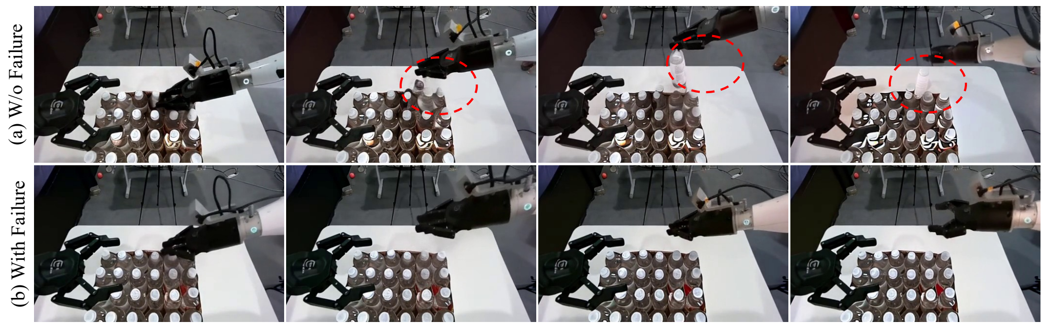


Figure 10: Impact of Failure Data on Trajectory Generation. Without failure data: The model overfits to success-only trajectories, incorrectly 'hallucinating' that the bottle has been grasped by the robotic arm.

The primary challenge lies in distinguishing between upward tossing and upward shaking, as they exhibit fundamentally different acceleration profiles. Upward tossing involves a sharp, high-acceleration movement, whereas upward shaking follows a smoother, low-acceleration trajectory. Without the Delta Action Module, spatial-aware action recognition models often fail to differentiate between these motions, resulting in incorrect predictions. This leads to temporal inconsistency, such as flickering or the sudden disappearance of objects (e.g., the ham) due to erratic motion transitions, as highlighted by the dashed red boxes in Figure 11.

The Delta Action Module addresses these limitations by introducing acceleration-aware action decomposition. By explicitly modeling the time-derivative of actions, the module captures second-order dynamics (velocity changes), enabling it to differentiate between high-acceleration motions like tossing and low-acceleration motions like shaking. As a result, the Delta Action Module ensures significantly stronger motion consistency and reduces temporal errors compared to configurations without it.

Figure 11: Results of generated videos under identical conditions with and without the Delta Action Module. Red boxes highlight regions with inconsistent or hallucinated results.

5 Conclusion

In this paper, we introduced EVAC, an embodied world model with action-conditioned capabilities. We proposed multi-level action condition injection strategies and utilized camera ray maps to model dynamic camera's motion. Through extensive experiments, we demonstrated the dual functionality of EVAC: serving as both a data engine for policy learning and an evaluator for trained policy models. However, limitations remain: our gripper representation may not generalize to complex end-effectors, and potential applications like actor-critic integration remain unexplored.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, and etc. Rt-1: Robotics transformer for real-world control at scale, 2023.

[3] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[4] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[5] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[6] Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation, 2025.

[7] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2021.

[8] Ruiqi Gao, Aleksander Hoł yński, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pages 75468–75494, 2024.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[10] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation, 2024.

[11] Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Yue Liao, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse: Envisioning embodied future space for robotics manipulation, 2025.

[12] Siyuan Huang, Yue Liao, Siyuan Feng, Shu Jiang, Si Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Adversarial data collection: Human-collaborative perturbations for efficient and robust robotic imitation learning. *arXiv preprint arXiv:2503.11646*, 2025.

[13] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[14] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[15] Xiaoqi Li, Lingyun Xu, Jiaming Liu, Mingxu Zhang, Jiahui Xu, Siyuan Huang, Iaroslav Ponomarenko, Yan Shen, Shanghang Zhang, and Hao Dong. Crayonrobo: Toward generic robot manipulation via crayon visual prompting. *arXiv preprint arXiv:2505.02166*, 2025.

[16] Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. *arXiv preprint arXiv:2403.20271*, 2024.

[17] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[18] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Siran Chen, Ying Shan, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos, 2024.

[19] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[20] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019.

[21] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya 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.

[22] Physical Intelligence. Pi0: [title of the blog post]. `https://www.physicalintelligence.company/blog/pi0`, 2025. Accessed: 2025-04-23.

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021.

[25] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[26] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[27] Boyang Wang, Nikhil Sridhar, Chao Feng, Mark Van der Merwe, Adam Fishman, Nima Fazeli, and Jeong Joon Park. This&that: Language-gesture controlled video generation for robot planning, 2024.

[28] Tan Wang, Linjie Li, Kevin Lin, Yuanhao Zhai, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for realistic human dance generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9326–9336, 2024.

[29] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems*, 36:7594–7611, 2023.

[30] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[31] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023.

[32] Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators, 2024.

[33] Xiaoyu Zhang, Matthew Chang, Pranav Kumar, and Saurabh Gupta. Diffusion meets dagger: Supercharging eye-in-hand imitation learning. *arXiv preprint arXiv:2402.17768*, 2024.

[34] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and Introduction Section, we clearly demonstrate the contribution and scope of the proposed EVAC.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the main text, **Conclusion** Section, (5), we outlined the limitations related to our work, hoping to guide more future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have described the relevant details in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provided the GitHub link in our project page.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 4.1 and App. A, we clearly demonstrated experimental settings, including model architecture, training settings, evaluation settings, etc.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the high cost of training, we do not repeat the same experiments. But we perform multiple independent runs with different random seeds for the policy evaluation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 4.1 and Appendix A, we have provided sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We guarantee that the research conducted in the paper complies with NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper has no negative societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our license name is CC-BY 4.0. We will cite the original works and properly acknowledge the authors of any existing assets (e.g., code, models, datasets) used in our paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

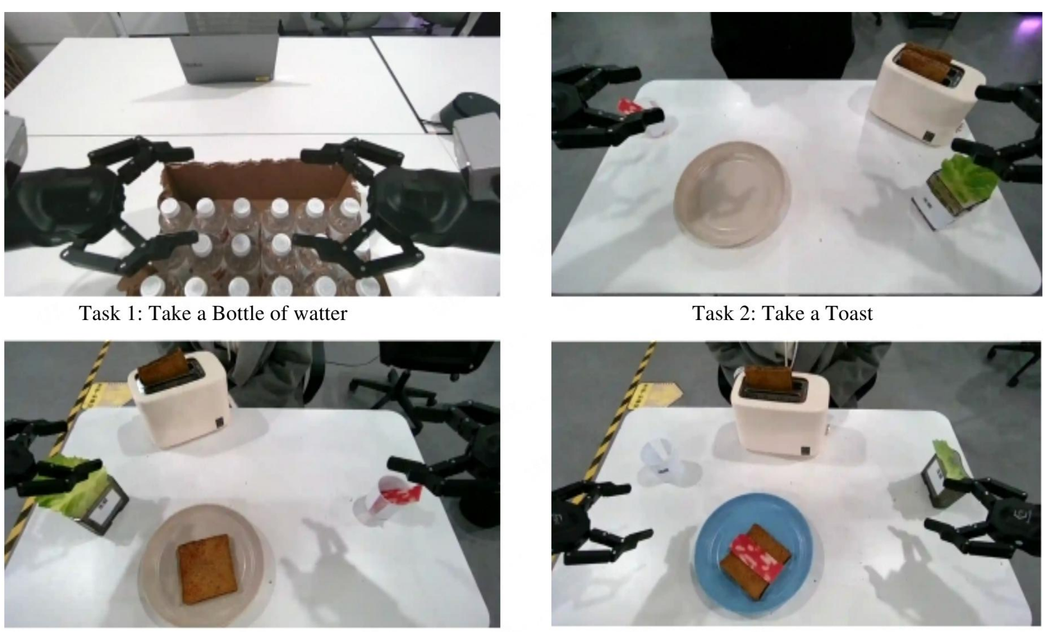
A Model Parameters and Training Configuration

The hyperparameters of the model architecture and training setup are provided in Table 2.

Table 2: Model Parameters and Training Configuration. **F** indicates the fisheye camera.

Category	**Configuration**
Diffusion Parameters	
Steps / Noise schedule	1000 / Linear
β_0 / β_T	0.00085 / 0.0120
UNet Architecture	
Input channels (total)	19 (With Latent Image: 4, Condition Latent: 4, Action:4, Ray Map:6, Dropout Mask:1)
z-shape / Base channels	$40 \times 64 \times 4$ / 320
Attention resolutions	1,2,4
Blocks per resolution / Context Dim	2 / 1024
Data Configuration	
Video resolution / Chunk size	320×512 / 16
Views	head(with 2 **F**), 2 wrists
Training Setup	
Learning rate / Optimizer	5×10^{-5} / Adam
Batch size per GPU	8 (Single-) / 1 (Multi-view)
Parameterization / Max steps	v-prediction / 100,000
Gradient clipping	0.5 (norm)

B Evaluation Tasks Visualization



Task 1: Take a Bottle of watter

Task 2: Take a Toast

Task 3: Take a Bacon

Task 4: Take a Lettuce

Figure 12: Initial conditions for the four manipulation tasks used in policy evaluation experiments.

C Precise Trajectory-following Ability

We present results generated under the same initial conditions but with different trajectories in Figure 13.

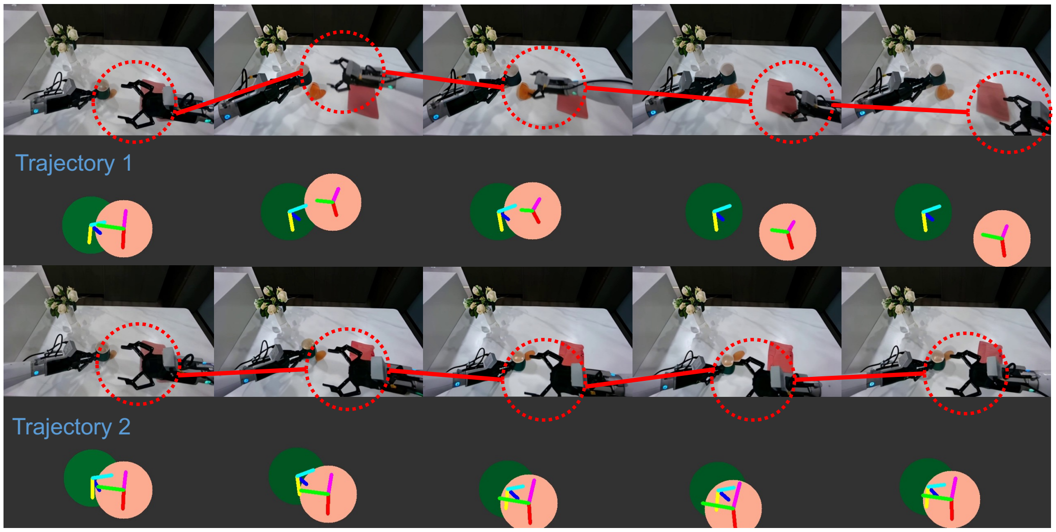


Figure 13: Results with the same initial condition but condition on different trajectories.

D Robustness to Calibration & Action Noise.

We train and evaluate EVAC on robots of the same robot type but different units, introducing natural extrinsic calibration offsets. But we skip any re-calibration and use factory-default parameters. Despite this, EVAC still achieves high evaluation consistency. Operators confirm alignment by inspecting overlaid action maps on the live camera feed (Figure 3) and report no misalignments, indicating that neither extrinsic offsets nor minor action jitter compromise EVAC's evaluation fidelity.