

Scalable Policy Evaluation with Video World Models

Wei-Cheng Tseng^{1,2,3}, Jinwei Gu¹, Qinsheng Zhang¹, Hanzi Mao¹, Ming-Yu Liu¹
Florian Shkurti^{2,3}, Yen-Chen Lin¹

Abstract—Training generalist policies for robotic manipulation has shown great promise, as they enable language-conditioned, multi-task behaviors across diverse scenarios. However, evaluating these policies remains difficult because real-world testing is expensive, time-consuming, and labor-intensive. It also requires frequent environment resets and carries safety risks when deploying unproven policies on physical robots. Manually creating and populating simulation environments with assets for robotic manipulation has not addressed these issues, primarily due to the significant engineering effort required and the often substantial sim-to-real gap, both in terms of physics and rendering. In this paper, we explore the use of action-conditional video generation models as a scalable way to learn world models for policy evaluation. We demonstrate how to incorporate action conditioning into existing pre-trained video generation models. This allows leveraging internet-scale in-the-wild online videos during the pre-training stage, and alleviates the need for a large dataset of paired video-action data, which is expensive to collect for robotic manipulation. Our paper examines the effect of dataset diversity, pre-trained weight and common failure cases for the proposed evaluation pipeline. Our experiments demonstrate that, across various metrics, including policy ranking and the correlation between actual policy values and predicted policy values, these models offer a promising approach for evaluating policies without requiring real-world interactions.

I. INTRODUCTION

Robot learning has seen remarkable strides in the development of generalist robot manipulation policies that are capable of understanding natural language instructions and performing diverse tasks across different environments and robot embodiments [1], [2], [3], [4]. However, as these policies and the tasks they support grow in diversity and complexity, a fundamental challenge arises: how do we systematically evaluate the capabilities of generalist robot manipulation policies in a scalable and efficient way?

Current evaluation methodologies face significant obstacles that hinder rapid development and scalable deployment. Real-world testing and policy rollouts are the gold standard for estimating the performance of a policy, but they often incur high data collection and annotation costs, extensive robotic infrastructure requirements, limited reproducibility [5], and are severely limited by the need for environment resets. On the other hand, simulation-based approaches for policy evaluation require substantial manual effort to mitigate the physics and rendering sim-to-reality gap for robotic manipulation [6], [7]. Constructing realistic simulations demands careful 3D modeling [8], reward engineering [9], and domain randomization [10], yet these efforts often fail to bridge the sim-to-real gap in terms of physics accuracy and visual fidelity.

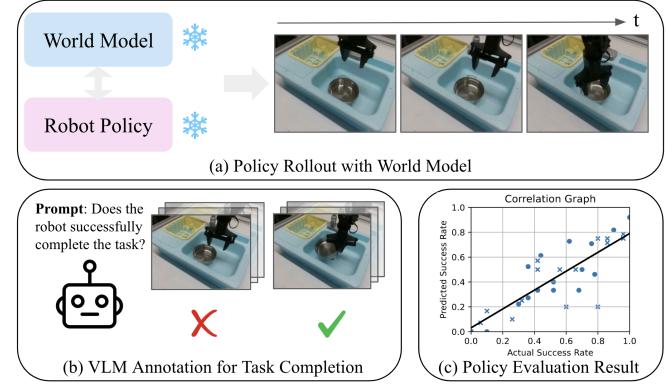


Fig. 1: **Overview**. We propose using action-conditional video prediction models as world model for scalable policy evaluation. The process involves: (a) deploying a policy into the video model to generate rollout videos conditioned on action sequences and initial frame, (b) using a VLM to judge task success or failure, and (c) computing the correlation between predicted and actual policy performance to evaluate the reliability of the approach.

Recently, action-conditional video prediction models [11], [12], [13], [14], [15], [16] have emerged as a promising tool for robotics to turn video prediction models into world models that can be used for robotics. These models learn to predict future frames given current observations and the agent’s actions, providing a data-driven simulation framework for evaluating robot policies. Compared to existing methods, this approach provides several key advantages: (1) unlike real-world testing, this method ensures reproducibility and does not require expensive physical equipment setups; (2) unlike handcrafted simulations, these models learn directly from real-world data, capturing complex visual and dynamic properties; (3) instead of manually constructing virtual environments to align the appearance of the environment and objects, these models condition directly on real observations, enabling effective real-to-sim transfer; (4) model performance improves with larger datasets and models, without requiring extensive human effort to engineer new tasks.

To this end, we propose a scalable, fully automated framework for policy evaluation that leverages video models. Our framework enables the policy to interact with an action-conditional video prediction model [17], [18], [19], [20] to generate rollout videos. We then use an off-the-shelf Vision-Language-Model (VLM) [21], [22] to annotate these rollouts and assess task completion, eliminating the need for human intervention and assessment. We present the

¹Nvidia Research, ²University of Toronto, ³Vector Institute

correlation between the real policy performance and estimated performance from the action-conditional video model, using both simulation and real-world environments and tasks. Also, we systematically investigate the design space of diffusion-based action-conditional video prediction models and their effectiveness in evaluating robot policies. First, we examine how expanding the demonstration dataset (from 0.3 million to 1.1 million transitions) affects performance. While larger datasets improve generalization in terms of action-following, models still fail to enforce basic physical constraints, such as object permanence and rigid-body interactions. Second, we study how pre-training on large-scale text-to-video and image-to-video datasets affects generalization. Our results demonstrate that pre-trained models yield more physically plausible predictions, underscoring the promise of large-scale pre-training for building data-driven simulators. Third, we present experimental results using different pre-trained video model backbones. Finally, we highlight common failure cases in video models that may affect the reliability of policy evaluation. Our contributions can be summarized as follows:

- We systematically evaluate action-conditional video prediction models for robot policy assessment, identifying their strengths and key failure modes, particularly in out-of-distribution action handling and physical realism.
- We conduct an analysis of the effects dataset diversity, and pre-training of action-conditional video prediction models for policy evaluation tasks.
- We plan to release the proposed policy evaluation pipeline, including the initial frames for both real-world and synthetic tasks, the action-conditioned model, and the VLM for task completion to facilitate further benchmarking of action-conditioned video models in policy evaluation.

Supplementary materials, including additional qualitative results and an appendix are available on the project website.

II. RELATED WORK

Video Generation Models. Recent advancements in diffusion and autoregressive models have significantly improved video generation capabilities [23], [24], [25], [26], [17], [19]. These models can generate videos conditioned on diverse input modalities such as text, images, depth maps, segmentation masks, camera poses, and motion trajectories [27], [28]. Building on top of this progress, a growing body of work explores how video models conditioned on actions can function as interactive simulators, supporting applications in entertainment [14], [29], [30], planning [31], and reinforcement learning [11], [12]. This progress has spurred interest in their potential integration into embodied AI systems. In this work, we systematically investigate the impact of scaling model size, dataset diversity, and pre-training tasks on the effectiveness of using action-conditioned video models as a data-driven simulator for robot policy evaluation.

Robot Policy Evaluation. Reliable robot policy evaluation is essential for benchmarking algorithmic progress and ensuring transparency in real-world performance. In autonomous driving, policies are commonly validated in simulation

across diverse scenarios before real-world deployment [32], [33]. However, simulating dynamic interactions with objects, realistic materials, and complex lighting conditions in robotic manipulation remains an open challenge. Efforts such as YCB [34] have standardized objects for evaluation, yet standardizing the remaining factors within an environment remains a challenge. Real-world robot challenges, including the Amazon Picking Challenge [35] and the DARPA Robotics Challenge [36], address this by using fixed physical test sites. However, these setups demand extensive preparation and maintenance. These challenges motivate an increasing interest in using simulation to estimate policies' success rates. SIMPLER [7] constructs physical simulators to approximate real-world environments. They show that one can deploy the policies trained with real-world data in the simulator to estimate their performance in the real world. However, this approach has two major limitations: (1) the reliance on extensive domain randomization to close the sim-to-real gap, and (2) the need for manually curated 3D data and precise environment design. In contrast, our approach eliminates manual engineering by leveraging action-conditional video prediction models to generate simulation environments directly from policies' observations, offering a scalable and data-driven alternative to classical simulator-based evaluation methods. AutoEval [37] presents a system that autonomously evaluates generalist robot policies in the real world. However, real-world policy evaluation still faces several challenges, such as motor overheating and long wall-clock times. Moreover, real-world evaluations are inherently imperfect and subject to noise and variability. WorldEval [38] proposed using a video model for policy evaluation; however, it requires the video model to utilize features from the action decoder as conditioning signals, which limits its applicability for evaluating new policies.

III. METHOD

A. Evaluating Robot Policies

Let N be the total number of policies. Let π_i and π_j be a pair of them, with real-world cumulative performance R_i and R_j , for example, their average success rates across a representative set of tasks, as estimated by real-world rollouts. Our goal is to construct a world simulator S , for which there is a strong correlation between the relative performances R_i and R_j in the real world and the relative performances in simulation $R_{S,i}$ and $R_{S,j}$.

Metrics. To evaluate our approach, we use the metrics proposed by SIMPLER [7], including MMRV and the Pearson correlation coefficient [40]. Specifically, MMRV (Mean Maximum Rank Violation) is defined as follows:

$$\text{RankViolation}(i, j) = |R_i - R_j| \cdot \mathbf{1} \left[(R_{S,i} < R_{S,j}) \neq (R_i < R_j) \right] \quad (1)$$

$$\text{MMRV}(R, R_S) = \frac{1}{N} \sum_{i=1}^N \max_{1 \leq j \leq N} \text{RankViolation}(i, j) \quad (2)$$

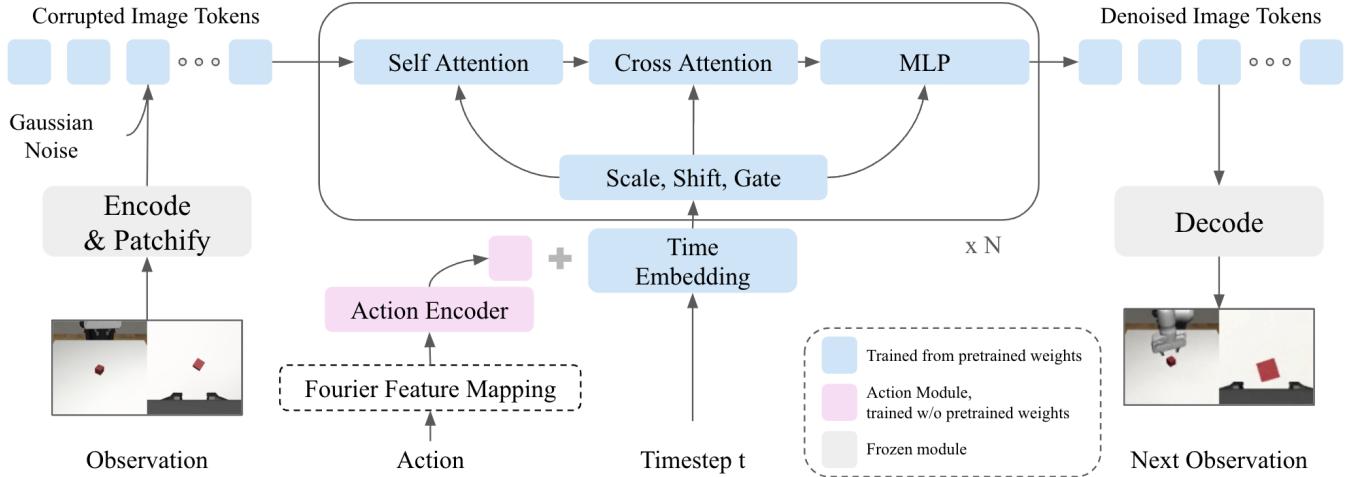


Fig. 2: **Network Architecture for Action-Conditional Video Prediction with a DiT-Based Video Model such as Cosmos-Predict2-2B [18].** We process the raw action information with Fourier feature mappings [39] and encode it into a latent action token. The latent action token is added to a time-embedding that enables us to make the video diffusion model action-conditional.

The key concept underlying MMRV is *rank violation* between two policies, π_i and π_j , which quantifies the extent to which the world model misranks the policies relative to their true performance margins in the real world. MMRV aggregates the N^2 rank violations by computing the average worst-case rank violation for each policy.

Once we have a world model S , we can roll out the policy autoregressively by allowing it to interact with the world model. After the completion of a model-based rollout, we leverage Vision-Language Models [41], [42] to assess whether the policy successfully completes the task. To be more specific, we subsample frames from the rollout video and feed them into VLM, a prompt is also provided to identify the definition of task completion. For the detailed prompting design of the VLM, we refer the reader to the supplemental material on the project website.

B. Learning Action-conditional Video Prediction Models

Let \mathcal{O} be the space of observations (e.g., video frames), and let \mathcal{A} be the space of actions, which is the end-effector pose of the robot arm in world coordinates. At each time step t , a policy π receives a sequence of past observations $o_{1:t} = \{o_1, \dots, o_t\} \in \mathcal{O}$ and decides on an action $a_{1:t} \in \mathcal{A}$. This chosen action is then executed in the environment, leading to the observation of the next step $o_{t+1} \in \mathcal{O}$.

Our goal is to build an action-conditional video prediction model that forecasts o_{t+1} given $o_{1:t}$ and $a_{1:t}$. Formally, we aim to learn the *dynamics* of the environment $p(o_{t+1} | o_{1:t}, a_{1:t})$. This approach is task-agnostic, as multiple tasks often share the same observation space (video frames). Consequently, the learned model can leverage data from multiple tasks, potentially improving performance through knowledge transfer. Moreover, it provides a consistent simulator that can be applied broadly across tasks. By iteratively sampling from this predictive model—each time conditioning on its own previously generated observations—we can simulate a complete interaction trajectories.

Video Model Architectures and Implementation. We leverage the open-sourced pre-trained video diffusion model, Cosmos-Predict2-Video2World-2B [17] (hereafter referred to as Cosmos), as the backbone of our action-conditional video prediction models. Cosmos-Predict2-Video2World-2B is a Diffusion Transformer [43], but it has additional cross-attention layers between self-attention and feedforward layers for conditioning. It is pre-trained with text-to-video and image-to-video on large-scale dataset. In total, it has 2B trainable parameters. Cosmos-series models, like many other video generation models, are not designed by default to condition on actions. On one hand, this has the benefit of allowing learning from in-the-wild video data online; on the other hand, this is problematic for robotics simulation, which requires conditioning on actions. To augment the model with action-conditioning capability, we add action embeddings to the time embeddings. The action embeddings are generated by first applying Fourier feature mappings [39] to the raw action signals to enable the network to learn high-frequency functions from low-dimensional inputs, followed by processing them through a MLP. The FLOPs of this MLP are significantly smaller than the rest of the network, so the modification doesn't affect the inference and training efficiency. The model architecture is summarized in Figure 2, with additional details in appendix VII. The entire framework is trained end-to-end using the EDM objective [44], [45], following the same setup as Cosmos [17], [18].

C. Policy Rollout Data Augmentation

To train the video model we use demonstration trajectories collected by human teleoperation. However, the action distribution within the demonstration trajectories could be limited, for instance, the trajectories in the demonstration dataset typically correspond to successful trajectories, and failures are rare. To address this, we augment the demonstration dataset with rollouts from policies with different manipulation performance to enhance the generalization capability of the

TABLE I: Quantitative Results for Policy Evaluation in Synthetic Setting. Pearson correlation coefficient (higher is better) and MMRV (Mean Metric Rank Violation; lower is better) between video model and simulator policy evaluations.

World Model	Lift		Can		Square		Tool Hang	
	Pearson \uparrow	MMRV \downarrow						
Cosmos-Predict2-2B (Ours)	0.879	0.015	0.860	0.141	0.847	0.165	0.833	0.217
w/o Policy Rollouts	0.856	0.222	0.744	0.226	0.505	0.384	0.502	0.288
w/o Pre-trained Weights	0.822	0.220	0.716	0.233	0.523	0.245	0.645	0.363
Cosmos-Predict-1-7B	0.845	0.188	0.750	0.231	0.606	0.204	0.500	0.281
Cosmos-Predict-1-14B	0.800	0.260	0.716	0.233	0.511	0.261	0.666	0.360
Wan-2.1-14B-LoRA	0.842	0.180	0.746	0.166	0.801	0.179	0.789	0.244
IRASim	0.835	0.103	0.780	0.230	0.213	0.708	0.642	0.330

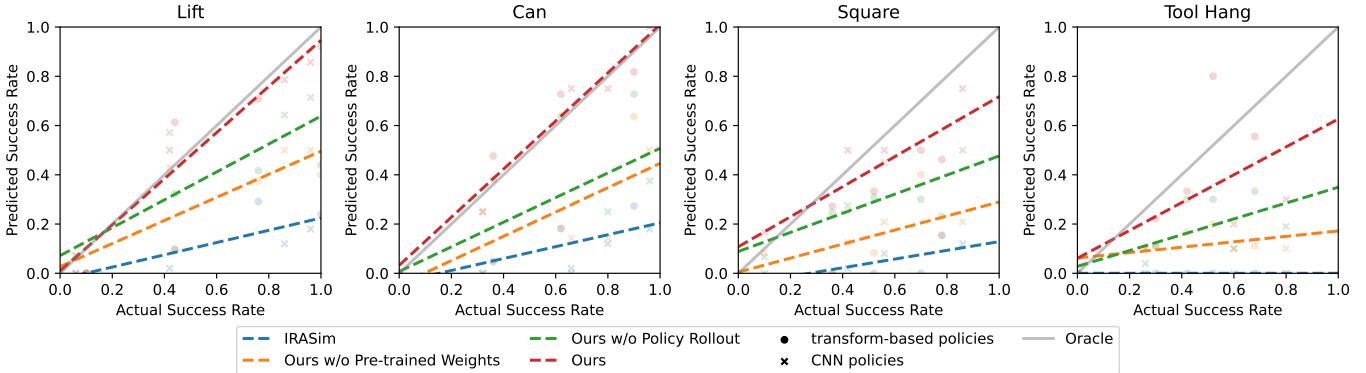


Fig. 3: Correlation Plots for Policy Evaluation. We plot the correlation between predicted policy performance via evaluation in the video model vs. actual policy performance for the simulation tasks.

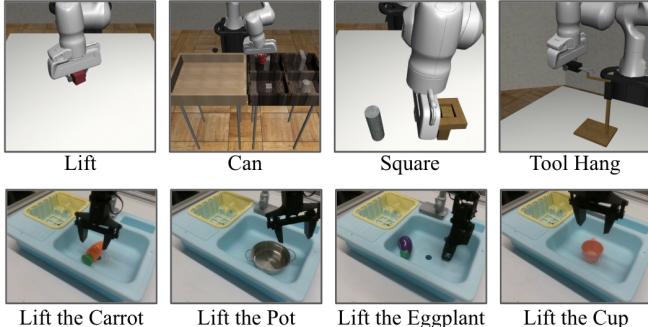


Fig. 4: Manipulation Tasks. We evaluate the proposed approach in both synthetic and real-world tasks. In the synthetic setting, we mainly consider four tasks in RoboMimic [46]: lift, can, square and tool hang. For real-world experiments, we consider four tasks from the Bridge benchmark: lifting a carrot, a pot, an eggplant, and a cup.

action-conditional video prediction model.

IV. EXPERIMENTS

A. Synthetic Setting

For controllable study , we first test our policy evaluation paradigm in a synthetic setting , where the environment dynamics are treated as ground truth . In this case, dynamics in synthetic environment are treated as ground-truth dynamics. Specifically, we ask whether the video model can predict the performance ranking of policies trained entirely on synthetic data and evaluated in the same synthetic environment.

We consider the following four tasks in RoboMimic [46]:

- **Lift:** The robot lifts a small cube. This is the simplest task.
- **Can:** The robot moves a Coke can from a large bin to a smaller target bin. It is harder than Lift because grasping the can is less stable, and precise placement is required.
- **Square:** The robot places a square nut onto a rod. This is more challenging than Lift and Can, as it requires precise grasping and careful alignment.
- **Tool Hang:** The robot assembles a frame by inserting a hook into a base and then hanging a wrench on the hook. This is the most complex task, involving multiple stages of precise, dexterous, and rotation-heavy movements.

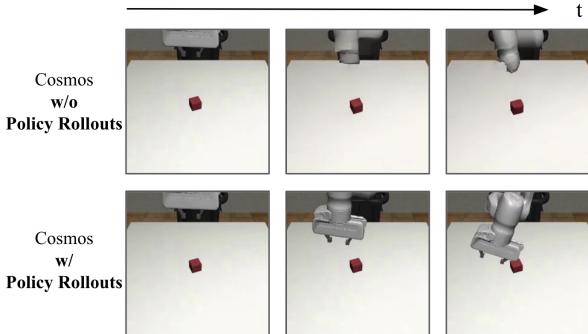
We train action-conditional video prediction models using both human demonstration trajectories and policy rollouts for four tasks from the RoboMimic [46] simulation environment. The original RoboMimic dataset contains 1,700 successful demonstrations. To expand this dataset, we add rollouts from transformer-based policies trained in Robosuite [47]. These rollouts introduce more diverse behaviors than human demonstrations¹. A summary of the augmented dataset is provided in Appendix VIII . The combined dataset is split into training, validation, and test sets.

For evaluation, we measure video prediction quality with PSNR [48], SSIM [49], latent L2, and FVD [50]. Latent L2 and PSNR capture reconstruction error in latent and pixel

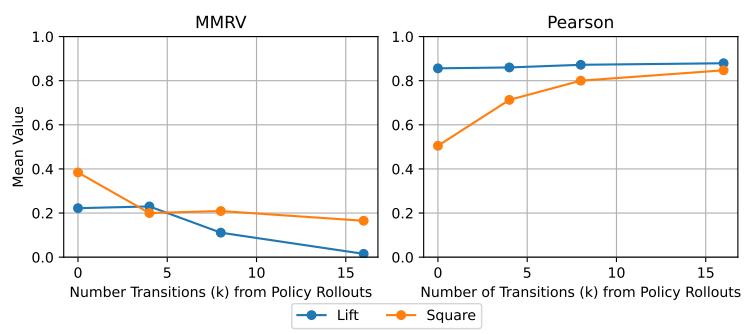
¹Importantly, the policies used to generate rollouts for training are not reused in later policy evaluation experiments.

TABLE II: **Quantitative Results for Action-Conditional Video Prediction.** We generate 64 future frames in an auto-regressive manner and evaluate on both RoboMimic (in-distribution) and policy rollout (out-of-distribution) trajectories.

World Model	RoboMimic (In-Distribution)				Policy Rollouts (Out-of-Distribution)			
	PSNR \uparrow	SSIM \uparrow	Latent L2 \downarrow	FVD \downarrow	PSNR \uparrow	SSIM \uparrow	Latent L2 \downarrow	FVD \downarrow
Cosmos-Predict2-2B	20.622	0.868	0.315	98.455	15.996	0.793	0.458	250.234
w/o Policy Rollouts	18.712	0.841	0.362	130.211	14.931	0.783	0.461	285.001
w/o Pre-trained Weights	17.731	0.834	0.384	127.695	14.754	0.766	0.519	291.158



(a) Comparison between with and without policy rollouts.



(b) Effect of Policy Rollouts on Policy Evaluation

Fig. 5: **Effect of Policy Rollouts on Policy Evaluation.** (a) Without policy rollouts, predictions degrade over time—for example, the gripper gradually disappears. Incorporating rollouts significantly improves visual quality and realism. (b) Increasing the number of policy rollouts in the video model’s training set consistently enhances policy evaluation performance.

space, respectively. SSIM measures similarity in brightness, contrast, and structure, while FVD compares feature distributions to assess overall video quality.

For policy evaluation, we train a set of policies using diffusion policy [51] with either CNN- or transformer-based architectures on RoboMimic [46] data. Each policy takes two image inputs (a gripper view and an agent view) from the most recent two timesteps and predicts the next eight actions. To create policies of varying quality, we train them for different numbers of steps.

Table I presents the quantitative results of evaluating these policies with the Cosmos model, and Figure 3 shows the corresponding correlation plots. Overall, our approach achieves sufficient accuracy to provide reliable policy ranking signals. That said, correlation is generally weaker for long-horizon tasks (e.g., Tool Hang and Square) than for shorter ones (e.g., Lift and Can). Video prediction performance is reported separately in Table II.

B. Ablation Study

To better understand the factors that affect video models for policy evaluation, we study the role of training data, pre-trained weights, and different model initializations.

1) *Effectiveness of Data Augmentation:* We investigate whether adding policy rollouts to the training set improves the video diffusion model. For comparison, we also train a baseline model using only human demonstration trajectories.

As shown in Table II, adding policy rollouts—trajectories generated by executing policies in RoboMimic—improves video prediction performance, especially on out-of-distribution (OOD) data. Figure 5a further shows that models trained without rollouts produce more frequent

artifacts. For policy evaluation, excluding rollouts leads to a sharp drop in the correlation between actual and predicted policy performance (Figure 3, Table I), which we attribute to lower-quality video predictions when interacting with policies.

We also analyze how the number of policy rollouts affects evaluation on the Lift and Square tasks (Figure 5b). In general, more rollouts improve evaluation accuracy, with the effect being more pronounced for the harder Square task than for the simpler Lift task.

2) *Effectiveness of Pretrained Weights:* To assess the impact of pretraining, we compare models trained from scratch against those initialized with the pre-trained weights of Cosmos-Predict2-Video2World-2B [17], which was trained on a large, high-quality text-to-video dataset for physical-AI applications.

As shown in Table II, removing pretrained weights causes a substantial drop in video prediction quality for both in-distribution and out-of-distribution data. Models trained from scratch exhibit more frequent distortions and hallucinations compared to pretrained models.

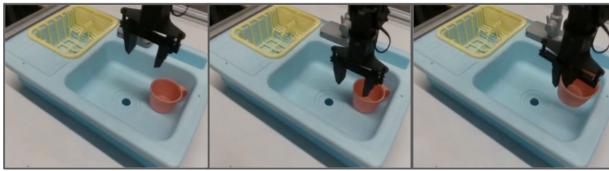
We hypothesize that pretraining provides useful priors about physical dynamics that are difficult to learn from demonstration data alone. Without these priors, models trained from scratch often display hallucination, deformation, and occlusion when the gripper interacts with objects or when objects collide (see Figure 7). This suggests that the model without pre-training lacks a robust understanding of physics, limiting its effectiveness for policy evaluation.

Policy evaluation results, summarized in Figure 3 and Table I, confirm that pretrained priors significantly improve

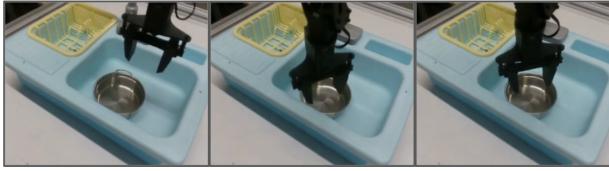
Initial Observation



Rollout Frames

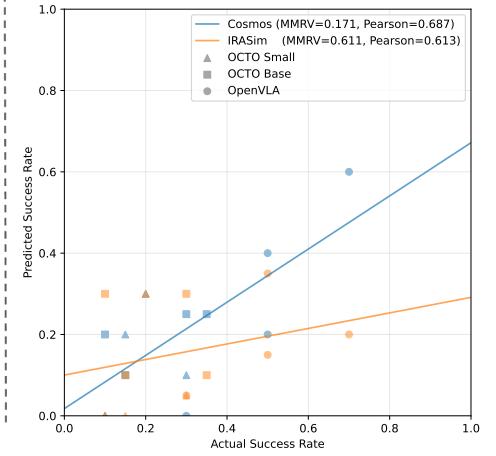


Task: Lift the cup ✓



Task: Lift the pot ✗

(a) Qualitative Result for Policy Rollout on Video Model



(b) Correlation Graph

Fig. 6: **Policy Evaluation on the Bridge Dataset [52], [53]** (a) We roll out the policy Octo-Small [3] with the action-conditional video model, trained on the Bridge dataset. The top row shows a successful lift of the cup, while the bottom row illustrates a failure rollout where the policy is unable to lift the pot. (b) Our video model can serve as an evaluation environment, and the performance acquired from it has positive correlation with real-world performance.

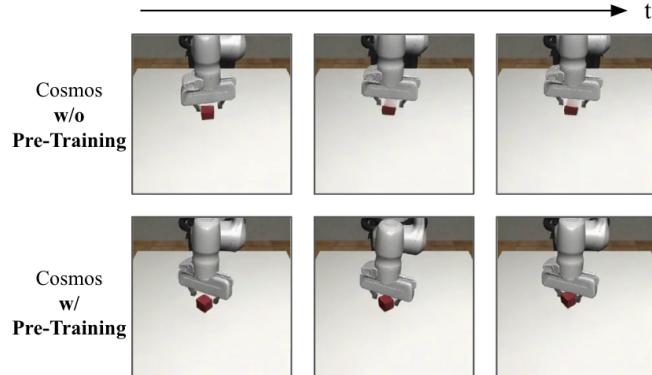


Fig. 7: Pretrained weights mitigate object disappearance during interaction, unlike randomly initialized weights.

evaluation accuracy.

3) *Different Pre-trained Models*: We benchmark policy evaluation across several video model backbones, including Wan2.1 [19] fine-tuned with LoRA, Cosmos-Predict1 [17], and IRASim [20]. Among these, Cosmos-Predict2—pre-trained on physical-AI data—consistently achieves the best performance. Its advantage likely comes from the strong alignment between its pretraining dataset and the physical-AI domain, which enables better generalization to robotics tasks compared to Wan2.1, whose pretraining was on general video data.

In contrast, IRASim, which is trained entirely from scratch without pretraining, performs the worst. This further underscores the importance of pre-trained weights for effective policy evaluation. Implementation details for post-training across different models are provided in Appendix VI-A.2 .

C. Policy Evaluation in Real-World Setup

We next examine whether our policy evaluation approach generalizes to real-world environments and generalist policies.

Following Simpler [7], we evaluate OCTO-Small, OCTO-Base [3], and OpenVLA [4] in the Bridge setup [52], [53]. The evaluated tasks include "lift the pot", "lift the carrot", "lift the eggplant", and "lift the cup".

For training the video model, we use the Bridge V2 dataset [53], [52]. To better align the original Bridge setup with our replicated setup, we additionally collect 300 trajectories in our own environment and include them in the training dataset.

As shown in Figure 6, our world model provides an effective evaluation environment: policy performance predicted by the model correlates positively with actual performance in the real world. Further details on the real-world experiments are provided in Appendix VI-A .

D. Failure Case Analysis

We have demonstrated the potential and feasibility of using video prediction models for policy evaluation. In this section, we discuss the limitations of our approach and highlight common failure modes observed in both simulation and real-world experiments. More quantitative analysis about failure case and capacity of the based model are provided in appendix.

Hallucinations. Even after post-training the video prediction model on our demonstration dataset to make it action-conditional and improve physical plausibility, we still observe problems such as inconsistent object behavior and lack of object permanence (Figure 8(a)). These artifacts can introduce bias into policy evaluation. In addition, because rollout videos are generated in an autoregressive manner, errors accumulate over time. This compounding effect becomes especially severe in long-horizon tasks—for example, each rollout of the Tool Hang task involves 700 transitions, where small prediction errors can quickly escalate.

Multi-View Inconsistency. Our world model takes two concatenated inputs (the gripper view and the agent view)



Fig. 8: Common Failure Cases in Policy Evaluation with Video Models (a) **Hallucination.** The first instance shows that the handle of the square disappears due to a hallucination. This would lead to failure of policy. The other instance shows that Cosmos will hallucinate the eggplant into the gripper, which could lead to overestimation of policy performance. (b) **Multi-View Inconsistency.** The generated result may fail to ensure multi-view consistency. For instance, in the agent view, the can appears to be grasped by the gripper, but it is missing in the gripper view.

and predicts the corresponding two views at the next timestep. However, the predicted views can be inconsistent [54], with the same object appearing differently across viewpoints (Figure 8(b)). This issue likely stems from the absence of an explicit view-consistency loss during training. As with hallucinations, the problem is amplified in the autoregressive setting, where small inconsistencies accumulate over time.

Replicated Prediction. When the gripper’s displacement is very small, the diffusion model often replicates the conditional image as the predicted image. This leads to a static environment, which hinders policy evaluation by preventing the policy from progressing toward achieving its task. The issue is more common in Tool Hang. Because insertion requires extremely precise, fine-grained motions, the gripper’s displacements are minimal, making replication more likely.

VLM Annotation Errors. Automating task completion judgments with VLM is imperfect. Often, VLM annotations achieve only 65–80% accuracy. A more detailed analysis of these annotation errors is provided in Appendix VI-A.5 .

V. CONCLUSION

We investigated how to adapt pre-trained video prediction models into action-conditional world models for scalable robot policy evaluation. By conditioning large video models on actions, our approach enables policy assessment without the need for costly real-world rollouts or heavily engineered simulation environments. Our analysis underscores three key factors for reliable evaluation: the value of diverse training datasets, the benefits of pretrained weights, and the failure modes that can arise during prediction.

Across both synthetic and real-world experiments, action-conditional video models accurately predict policy performance, achieving strong correlations with ground-truth success rates and supporting policy ranking. These results point to video-based world models as a promising foundation for data-driven evaluation pipelines—reducing reliance on expensive robotic infrastructure and accelerating progress toward generalist robotic policies.

REFERENCES

- [1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, “ π_0 : A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.
- [2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.
- [3] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, “Octo: An open-source generalist robot policy,” in *RSS*, Delft, Netherlands, 2024.
- [4] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, “Openvla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [5] H. Kress-Gazit, K. Hashimoto, N. Kuppuswamy, P. Shah, P. Horgan, G. Richardson, S. Feng, and B. Burchfiel, “Robot learning as an empirical science: Best practices for policy evaluation,” *arXiv preprint arXiv:2409.09491*, 2024.
- [6] N. Funk, C. Schaff, R. Madan, T. Yoneda, J. U. De Jesus, J. Watson, E. K. Gordon, F. Widmaier, S. Bauer, S. S. Srinivasa, *et al.*, “Benchmarking structured policies and policy optimization for real-world dexterous object manipulation,” *RAL*, vol. 7, no. 1, pp. 478–485, 2021.
- [7] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, S. Levine, J. Wu, C. Finn, H. Su, Q. Vuong, and T. Xiao, “Evaluating real-world robot manipulation policies in simulation,” *arXiv preprint arXiv:2405.05941*, 2024.
- [8] X. Li, J. Li, Z. Zhang, R. Zhang, F. Jia, T. Wang, H. Fan, K.-K. Tseng, and R. Wang, “Robogsim: A real2sim2real robotic gaussian splatting simulator,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.11839>
- [9] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, “Eureka: Human-level reward design via coding large language models,” *arXiv preprint arXiv:2310.12931*, 2023.
- [10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *IROS*. IEEE, 2017, pp. 23–30.
- [11] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel, “Learning interactive real-world simulators,” *arXiv preprint arXiv:2310.06114*, 2023.
- [12] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret, “Diffusion for world modeling: Visual details matter in atari,” in *NeurIPS*, 2024.
- [13] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter, “Diffusion models are real-time game engines,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.14837>
- [14] H. Che, X. He, Q. Liu, C. Jin, and H. Chen, “Gamegen-x: Interactive open-world game video generation,” *arXiv preprint arXiv:2411.00769*, 2024.
- [15] Decart, J. Quevedo, Q. McIntyre, S. Campbell, X. Chen, and R. Wachen, “Oasis: A universe in a transformer,” 2024. [Online]. Available: <https://oasis-model.github.io/>
- [16] M. Rigter, T. Gupta, A. Hilmkil, and C. Ma, “Avid: Adapting video diffusion models to world models,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.12822>

- [17] NVIDIA, :, N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, D. Dworakowski, J. Fan, M. Fenzi, F. Ferroni, S. Fidler, D. Fox, S. Ge, Y. Ge, J. Gu, S. Gururani, E. He, J. Huang, J. Huffman, P. Jannaty, J. Jin, S. W. Kim, G. Klár, G. Lam, S. Lan, L. Leal-Taixe, A. Li, Z. Li, C.-H. Lin, T.-Y. Lin, H. Ling, M.-Y. Liu, X. Liu, A. Luo, Q. Ma, H. Mao, K. Mo, A. Mousavian, S. Nah, S. Niverty, D. Page, D. Paschalidou, Z. Patel, L. Pavao, M. Ramezanali, F. Reda, X. Ren, V. R. N. Sabavat, E. Schmerling, S. Shi, B. Stefaniak, S. Tang, L. Tchapmi, P. Tredak, W.-C. Tseng, J. Varghese, H. Wang, H. Wang, H. Wang, T.-C. Wang, F. Wei, X. Wei, J. Z. Wu, J. Xu, W. Yang, L. Yen-Chen, X. Zeng, Y. Zeng, J. Zhang, Q. Zhang, Y. Zhang, Q. Zhao, and A. Zolkowski, “Cosmos world foundation model platform for physical ai,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.03575>
- [18] N. C. Team, “Cosmos-predict2: General-purpose world foundation models for physical ai,” <https://github.com/nvidia-cosmos/cosmos-predict2/>, Jun 2025, apache-2.0 license, available under NVIDIA Open Model License.
- [19] A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu, “Wan: Open and advanced large-scale video generative models,” *arXiv preprint arXiv:2503.20314*, 2025.
- [20] F. Zhu, H. Wu, S. Guo, Y. Liu, C. Cheang, and T. Kong, “Irasim: Learning interactive real-robot action simulators,” *arXiv:2406.12802*, 2024.
- [21] Gemini Team, “Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities,” *arXiv preprint arXiv:2507.06261*, 2025.
- [22] OpenAI, “Chatgpt (may 2025 version),” <https://chat.openai.com>, 2025, accessed: 2025-05-16.
- [23] ———, ““video generation models as world simulators”, Jan 2024. [Online]. Available: <https://openai.com/index/video-generation-models-as-world-simulators/>
- [24] B. Lin, Y. Ge, X. Cheng, Z. Li, B. Zhu, S. Wang, X. He, Y. Ye, S. Yuan, L. Chen, *et al.*, “Open-sora plan: Open-source large video generation model,” *arXiv preprint arXiv:2412.00131*, 2024.
- [25] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, *et al.*, “Cogvideox: Text-to-video diffusion models with an expert transformer,” *arXiv preprint arXiv:2408.06072*, 2024.
- [26] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang, *et al.*, “Movie gen: A cast of media foundation models,” *arXiv preprint arXiv:2410.13720*, 2024.
- [27] D. Geng, C. Herrmann, J. Hur, F. Cole, S. Zhang, T. Pfaff, T. Lopez-Guevara, C. Doersch, Y. Aytar, M. Rubinstein, C. Sun, O. Wang, A. Owens, and D. Sun, “Motion prompting: Controlling video generation with motion trajectories,” *arXiv preprint arXiv:2412.02700*, 2024.
- [28] H. He, Y. Xu, Y. Guo, G. Wetzstein, B. Dai, H. Li, and C. Yang, “Cameractrl: Enabling camera control for text-to-video generation,” *arXiv preprint arXiv:2404.02101*, 2024.
- [29] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, *et al.*, “Genie: Generative interactive environments,” in *ICML*, 2024.
- [30] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter, “Diffusion models are real-time game engines,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.14837>
- [31] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, *et al.*, “Video language planning,” *arXiv preprint arXiv:2310.10625*, 2023.
- [32] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *CoRL*, 2017.
- [33] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, “Unisim: A neural closed-loop sensor simulator,” in *CVPR*, 2023.
- [34] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, “The ycb object and model set: Towards common benchmarks for manipulation research,” in *ICRA*, 2015.
- [35] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, “Analysis and observations from the first amazon picking challenge,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2016.
- [36] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski, “The darpa robotics challenge finals: Results and perspectives,” *The DARPA robotics challenge finals: Humanoid robots to the rescue*, pp. 1–26, 2018.
- [37] Z. Zhou, P. Atreya, Y. L. Tan, K. Pertsch, and S. Levine, “Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world,” *arXiv preprint arXiv:2503.24278*, 2025.
- [38] Y. Li, Y. Zhu, J. Wen, C. Shen, and Y. Xu, “Worldeval: World model as real-world robot policies evaluator,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.19017>
- [39] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, “Fourier features let networks learn high frequency functions in low dimensional domains,” 2020. [Online]. Available: <https://arxiv.org/abs/2006.10739>
- [40] K. Pearson, “Vii. note on regression and inheritance in the case of two parents,” *proceedings of the royal society of London*, vol. 58, no. 347–352, pp. 240–242, 1895.
- [41] Y. J. Ma, J. Hejna, A. Wahid, C. Fu, D. Shah, J. Liang, Z. Xu, S. Kirmani, P. Xu, D. Driess, T. Xiao, J. Tompson, O. Bastani, D. Jayaraman, W. Yu, T. Zhang, D. Sadigh, and F. Xia, “Vision language models are in-context value learners,” in *preprint*, 2024.
- [42] L. Guan, Y. Zhou, D. Liu, Y. Zha, H. B. Amor, and S. Kambhampati, “Task success is not enough: Investigating the use of video-language models as behavior critics for catching undesirable agent behaviors,” in *First Conference on Language Modeling*, 2024. [Online]. Available: <https://openreview.net/forum?id=otKo4zFKmH>
- [43] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *ICCV*, 2023, pp. 4195–4205.
- [44] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models,” in *Proc. NeurIPS*, 2022.
- [45] T. Karras, M. Aittala, J. Lehtinen, J. Hellsten, T. Aila, and S. Laine, “Analyzing and improving the training dynamics of diffusion models,” 2024. [Online]. Available: <https://arxiv.org/abs/2312.02696>
- [46] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, “What matters in learning from offline human demonstrations for robot manipulation,” in *CoRL*, 2021.
- [47] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, A. Maddukuri, S. Nasiriany, and Y. Zhu, “robosuite: A modular simulation framework and benchmark for robot learning,” in *arXiv preprint arXiv:2009.12293*, 2020.
- [48] A. Horé and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *CVPR*, 2010.
- [49] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [50] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, “FVD: A new metric for video generation,” 2019. [Online]. Available: <https://openreview.net/forum?id=rylgEULtdN>
- [51] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *RSS*, 2023.
- [52] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine, “Bridgedata v2: A dataset for robot learning at scale,” in *CoRL*, 2023.
- [53] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, “Bridge data: Boosting generalization of robotic skills with cross-domain datasets,” *Corr*, vol. abs/2109.13396, 2021. [Online]. Available: <https://arxiv.org/abs/2109.13396>
- [54] M. Asim, C. Wewer, T. Wimmer, B. Schiele, and J. E. Lenssen, “Met3r: Measuring multi-view consistency in generated images,” 2024.