Sample-Efficient RL via World Foundation Models for Synthetic Demonstrations

Albert Lin [*1] **Aarya Sumuk** [*2]

Abstract

Reinforcement learning (RL) for robotic manipulation typically requires large amounts of interaction data or expert demonstrations, both of which are expensive to collect. We propose leveraging world foundation models (WFMs), trained on internet-scale video data, as data engines to improve RL sample efficiency. Our approach finetunes the Cosmos Predict 2 WFM on merely 10 text-annotated expert demonstrations collected in the `robosuite` simulator (Zhu et al., 2020) on the MimicGen `StackThree` task (Mandlekar et al., 2023) to generate synthetic videos of the task. We then use this artificial dataset to train a diffusion policy via behavior cloning and subsequently finetune the policy with residual RL. Since the quality of the synthetic policy data is extremely important, we test several data-labeling approaches. We evaluate the efficacy of our approach in a series of experiments that vary the amount and mixture of data coming from real and synthetic sources. We find that while WFMs exhibit remarkably impressive video generation capabilities, this does not necessarily translate to effective policy data due to the challenges of physical alignment. We experimentally identify several obstacles in the way of artificially generating useful policy data and propose new directions for future work. The code used in this project can be found in the GitHub repository at the link in the footnote.[1]

1. Introduction

Reinforcement learning (RL) has shown remarkable success in robotic manipulation, but practical deployment remains limited by the large amounts of interaction data or expert demonstrations required for training. Collecting such data is expensive: real-world robot interactions are slow and carry the risk of hardware damage, while expert demonstrations require skilled human operators. This motivates the search for methods that can achieve strong performance with fewer environment interactions and minimal expert data.

Recently, world foundation models (WFMs) have emerged as powerful tools for modeling visual dynamics (NVIDIA et al., 2025; Jang et al., 2025). Trained on internet-scale video datasets, these models learn rich representations of physical processes and can transfer to new domains with limited finetuning. This capability suggests a compelling hypothesis: *Can WFMs serve as data engines that improve the sample efficiency of RL by synthesizing additional demonstrations from only a few expert-quality demonstrations?*

We propose a pipeline that finetunes the Cosmos Predict 2 WFM (NVIDIA et al., 2025) on a small set of expert demonstrations from the `robosuite` simulator using the MimicGen `StackThree` task, then uses the finetuned model to generate synthetic video rollouts drawn from the expert distribution. A proprioception estimation model (PEM) recovers proprioception labels from these generated videos, and an inverse dynamics model (IDM) recovers action labels. These labels are combined to produce a synthetic "expert" demonstration dataset. A diffusion policy is then trained using the denoising score matching objective (Chi et al., 2025) and subsequently finetuned via residual RL in simulation.

Since the quality of the synthetic policy data is extremely important, we test several data-labeling approaches. We evaluate the efficacy of our approach in a series of experiments that vary the amount and mixture of data coming from real and synthetic sources. We find that while WFMs exhibit remarkably impressive video generation capabilities, this does not necessarily translate to effective policy data due to the challenges of physical alignment. We experimentally identify several obstacles in the way of artificially

[*]Equal contribution . Both authors were enrolled in CS234. [1]Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, USA [2]Department of Mechanical Engineering, Stanford University, Stanford, CA, USA. Correspondence to: Albert Lin <albertkl@stanford.edu>, Aarya Sumuk <asumuk@stanford.edu>.

[1]https://github.com/albertklin/CS234-Final-Project - Access has so far been granted to the GitHub user `emmabrunskill`.

generating useful policy data and propose new directions for future work.

Our main contributions are:

1. A complete pipeline for generating labeled synthetic demonstrations from a WFM, requiring only 10 expert demonstrations and 1000 cheaply collected random rollouts.
2. A two-stage labeling architecture, comprised of proprioception estimation followed by state-based inverse dynamics, that overcomes the distribution shift between training data (random rollouts) and synthetic expert-like videos.
3. An empirical evaluation showing that while WFMs generate visually impressive videos, translating this into effective policy data remains challenging due to physical alignment errors.

The code used in this project can be found in our GitHub repository;[1] access has so far been granted to the GitHub user `emmabrunskill`.

2. Related Work

World models for policy learning. World models learn predictive representations of environment dynamics and have been used for planning, data augmentation, and policy learning. Early work by Ha & Schmidhuber (Ha & Schmidhuber, 2018) demonstrated training policies entirely in learned "dreams," and DreamerV3 (Hafner et al., 2025) scales this to diverse domains via latent imagination. More recently, large-scale video generation models have been applied to robotics: UniPi (Du et al., 2023) uses text-conditioned video diffusion as a planner, extracting actions via inverse dynamics at test time. DreamGen (Jang et al., 2025) generates synthetic robot videos from a finetuned world model and recovers actions via a learned IDM, producing "neural trajectories" for policy training, which is the closest prior work to our approach. UniSim (Yang et al., 2024) learns an interactive simulator from diverse data for training RL policies with zero-shot real-world transfer, and DINO-WM (Zhou et al., 2025) builds world models in pretrained visual feature space for zero-shot planning. Most recently, Cosmos Policy (Kim et al., 2026) and DreamZero (Ye et al., 2026) jointly model video and actions within a single diffusion process, eliminating the need for post-hoc action labeling entirely. Our work uses a pretrained WFM (Cosmos-Predict2) not for planning or direct control, but as a *data engine*: we finetune it on 10 demonstrations to generate an offline dataset of 200 labeled synthetic demonstrations for downstream policy training. Our primary contribution relative to DreamGen is a two-stage labeling architecture (PEM + State IDM) that addresses the distribution shift between random training data and expert-like synthetic videos; our findings on the physical grounding gap in video-only WFMs also motivate the joint video-action approaches of Cosmos Policy and DreamZero.

Imitation learning and data efficiency. Behavioral cloning learns policies directly from expert demonstrations but is sensitive to dataset size and quality; Mandlekar et al. (Mandlekar et al., 2021) show that architecture and data processing choices significantly impact BC performance in robotic manipulation. Diffusion Policy (Chi et al., 2025) achieves strong results by formulating action prediction as iterative denoising but typically requires hundreds of demonstrations for reliable performance. MimicGen (Mandlekar et al., 2023) addresses data scarcity by programmatically generating new demonstrations from a small set of human demonstrations via object-centric subtask decomposition. RL finetuning can further improve BC-initialized policies. In our downstream control setup, rather than finetuning the full policy with gradient-based actor-critic updates, we use a residual action-correction policy optimized with the Cross-Entropy Method (CEM), which preserves the structure of the pretrained behavioral cloning policy while allowing task-specific online improvement. Our work pursues a complementary data augmentation strategy: rather than programmatic generation or online RL exploration, we use a WFM to synthesize visually realistic demonstrations that are then labeled for offline policy training.

3. Approach

We formulate the manipulation task as a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $T(s' \mid s, a)$ is the transition dynamics, $R(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

Our pipeline consists of five stages: (1) expert demonstration collection, (2) WFM finetuning and synthetic video generation, (3) visual feature extraction and proprioception estimation, (4) inverse dynamics labeling, and (5) policy training via behavioral cloning followed by residual RL finetuning. An overview of our pipeline is visualized in Figure 1.

3.1. Environment and Dataset Details

We use the `robosuite` simulator (Zhu et al., 2020) with the MimicGen `StackThree` task (Mandlekar et al., 2023), in which a Panda robot arm must stack three colored blocks in a specified order using operational space control (OSC). Each trajectory $\tau_i = \{(o_t, a_t)\}_{t=0}^{T_i}$ consists of observations o_t and 7-dimensional actions $a_t = (\Delta x, \Delta y, \Delta z, \Delta\text{roll}, \Delta\text{pitch}, \Delta\text{yaw}, \text{gripper})$. The observation o_t comprises an 84×84 RGB image I_t from the `agentview` camera and a 9-dimensional proprioception

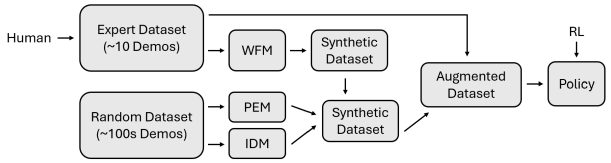


Figure 1. An illustration of our proposed sample-efficient RL pipeline, with shorthands for the world foundation model (WFM), inverse dynamics model (IDM), and proprioception estimation model (PEM).

vector p_t consisting of the end-effector position (3D), orientation quaternion (4D), and gripper finger positions (2D). We collect two datasets: (1) $\mathcal{D}_{\text{expert}}$, a set of 10 expert demonstrations from the MimicGen dataset (208-280 steps each), used exclusively for WFM finetuning; and (2) $\mathcal{D}_{\text{random}}$, a set of 1000 random-action rollouts (155 steps each), which provide paired (o_t, a_t) supervision for training the labeling models.

3.2. World Foundation Model Finetuning

We finetune the Cosmos Predict 2 WFM (NVIDIA et al., 2025), a 14B-parameter video generation model pretrained on internet-scale video data, to generate synthetic demonstrations of the `StackThree` task. We initialize from the GR00T-Dreams-DROID checkpoint (Bjorck et al., 2025), which was further pretrained on robotic manipulation videos from the DROID dataset, providing a strong prior for tabletop manipulation scenes. We apply low-rank adaptation (LoRA) with rank 32 to the attention and MLP layers of the denoiser network, keeping all pretrained weights frozen. The model is trained using the EDM denoising objective:

$$\mathcal{L}_{\text{WFM}}(D_\theta, \sigma) = \mathbb{E}_{\mathbf{x}_0,\, \mathbf{n}} \left[\| D_\theta(\mathbf{x}_0 + \mathbf{n};\ \sigma) - \mathbf{x}_0 \|_2^2 \right], \quad (1)$$

where D_θ is the denoiser, $\mathbf{x}_0$ is a clean video latent, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, and σ is sampled from a log-normal noise schedule. Only the 10 expert demonstrations are used for finetuning, each annotated with the text prompt: "the robot arm places the red block on the green block, then places the blue block on the red block." Training runs for 16,000 iterations with FusedAdamW (lr $\approx 4.6 \times 10^{-5}$, weight decay 0.2) and cosine learning rate decay with 100 warmup steps. Finetuning is distributed across 8 NVIDIA H100-80GB GPUs on a Google Cloud `a3-highgpu-8g` SPOT instance via FSDP with context parallelism, completing within 12 hours.

At inference time, the finetuned model is conditioned on a single first frame and the text prompt, then autoregressively generates 93-frame videos at 480×640 resolution and 16 fps using classifier-free guidance (scale 7) over 2 autoregressive steps. We generate 200 synthetic videos on the same cloud infrastructure, each conditioned on a different initial frame sampled from $\mathcal{D}_{\text{random}}$.

3.3. Visual Feature Extraction

To extract visual representations from video frames, we use a frozen DINOv3 (Siméoni et al., 2025) ViT-S/16 encoder (384-dimensional embeddings). We choose DINOv3 because its self-supervised pretraining produces spatially coherent features well-suited for dense prediction tasks, and its patch-level representations retain the spatial structure needed for localizing robot and object states from images. Rather than using only the CLS token, we extract the full grid of spatial patch tokens: at an input resolution of 112×144 pixels (after aspect-ratio-preserving resize from 480×640), this produces a 7×9 grid of 384-dimensional patch embeddings per frame. As we show in Section 4, spatial patch tokens dramatically outperform CLS tokens for both proprioception estimation (8.3×) and visual inverse dynamics (5.2×), as the spatial layout preserves fine-grained positional information about the robot and objects.

3.4. Proprioception Estimation Model

The synthetic videos contain only pixel observations; to recover the proprioception vector p_t required by the downstream policy, we train a proprioception estimation model (PEM) g_ψ by minimizing

$$\mathcal{L}_{\text{PEM}}(\psi) = \mathbb{E}_{(\mathbf{z}_t, p_t) \sim \mathcal{D}_{\text{random}}} \left[\| g_\psi(\mathbf{z}_t) - p_t \|^2 \right], \quad (2)$$

where $\mathbf{z}_t \in \mathbb{R}^{63 \times 384}$ are the spatial patch tokens extracted from frame I_t. The PEM is a transformer encoder-decoder (∼3.8M parameters): a linear projection maps the 384-dimensional patch tokens to a 256-dimensional hidden space, factored 2D positional encodings (separate learned row and column embeddings) are added, a 2-layer transformer encoder processes the sequence, and a single learned query attends to the encoder output via a cross-attention decoder to produce the 9-dimensional proprioception prediction. The model is trained on $\mathcal{D}_{\text{random}}$ using AdamW for 2000 epochs with cosine learning rate scheduling (∼4.8 hours on a single RTX 5090 GPU).

3.5. Inverse Dynamics Model

To recover action labels from the synthetic videos, we train an inverse dynamics model (IDM) that predicts the action taken between consecutive timesteps. A natural approach is a *visual IDM* that operates directly on visual embeddings; however, because $\mathcal{D}_{\text{random}}$ and the synthetic expert-like videos have very different visual distributions, we find that visual IDMs trained on random rollouts generalize poorly to expert-like trajectories (see Section 4).

Our solution is a *state IDM* f_ϕ that operates in proprioceptive

tion space, which is distribution-invariant:

$$\mathcal{L}_{\text{IDM}}(\phi) = \mathbb{E}_{(p_t, a_t, p_{t+1}) \sim \mathcal{D}_{\text{random}} \cup \mathcal{D}_{\text{expert}}} \left[\| f_\phi([p_t;\ p_{t+1}]) - a_t \|^2 \right] \tag{3}$$

The state IDM is a 4-layer MLP (hidden dimension 1024, ∼3.2M parameters) that takes the concatenation of two consecutive 9-dimensional proprioception vectors and predicts the 7-dimensional action. Because OSC actions are nearly a linear function of consecutive end-effector states, a pair of proprioception vectors is sufficient context. Crucially, the mapping from proprioception pairs to actions is distribution-invariant: the same physical relationship between consecutive states and the corresponding action holds regardless of whether the trajectory comes from a random or expert policy, eliminating the distribution shift that degrades visual IDMs. Training completes in under 1 hour on a single RTX 5090 GPU.

3.6. Synthetic Demonstration Labeling

The labeling pipeline combines the PEM and state IDM in a two-stage chain to produce fully labeled demonstrations from synthetic videos. For each synthetic video:

1. Decode 640×480 frames and extract DINOv3 spatial patch tokens (7×9×384 per frame).
2. Run the PEM on each frame's patch tokens to predict 9-dimensional proprioception.
3. Run the state IDM on consecutive predicted proprioception pairs to recover 7-dimensional actions.
4. Center-crop frames to 480×480 and resize to 84×84 to produce policy-compatible images.

The resulting observations (images + predicted proprioception) and actions are packed into a robomimic-format HDF5 dataset $\mathcal{D}_{\text{aug}}$, compatible with standard imitation learning pipelines. Labeling all 200 synthetic videos takes approximately 10 minutes on an RTX 5090 GPU.

3.7. Policy Training

We train a Diffusion Policy (Chi et al., 2025) via behavioral cloning on each training dataset variant, minimizing the action denoising objective:

$$\mathcal{L}_{\text{DP}}(\theta) = \mathbb{E}_{\boldsymbol{\epsilon}^k \sim \mathcal{N}(\mathbf{0},\, \sigma_k^2 \mathbf{I}),\, k} \left[\left\| \boldsymbol{\epsilon}^k - \boldsymbol{\epsilon}_\theta(\mathbf{o}_t,\, a_t^0 + \boldsymbol{\epsilon}^k,\, k) \right\|^2 \right], \tag{4}$$

where a_t^0 is the ground-truth action, $\boldsymbol{\epsilon}^k$ is noise with variance determined by the noise schedule at diffusion step k, $\mathbf{o}_t$ is the observation (image + proprioception), and $\boldsymbol{\epsilon}_\theta$ is the noise prediction network.

To evaluate the effect of synthetic demonstrations on downstream control, we train Diffusion Policies in `robomimic` under multiple dataset regimes: 200 expert demonstrations, 200 synthetic demonstrations, 50 expert + 150 synthetic, 100 expert + 100 synthetic, 10 expert demonstrations, and 10 expert + 100 synthetic demonstrations. All policies are trained for 1000 epochs on a single NVIDIA RTX 5050 GPU, with each run taking approximately 9 hours.

We then finetune selected BC-trained policies in simulation using a residual policy. Rather than updating the full diffusion policy, we freeze the pretrained policy and learn a small residual correction head that outputs an additive action offset from the current low-dimensional robot state:

$$a_t = \text{clip}\big(a_t^{\text{base}} + \alpha\, r_\phi(s_t),\ -1,\ 1\big), \tag{5}$$

where a_t^{base} is the action predicted by the frozen Diffusion Policy, s_t is the current low-dimensional observation, r_ϕ is the residual policy, and α is a small residual scaling factor. In our implementation, s_t consists of end-effector position, end-effector orientation quaternion, and gripper joint positions, and r_ϕ is a lightweight MLP with one hidden layer of width 32 and `tanh` nonlinearity.

We optimize the residual policy parameters with the Cross-Entropy Method (CEM), a population-based black-box optimization procedure. At iteration i, candidate parameter vectors are sampled from a Gaussian search distribution

$$w_j^{(i)} \sim \mathcal{N}\Big(\mu^{(i)}, \text{diag}\Big((\sigma^{(i)})^2\Big)\Big), \qquad j = 1, \ldots, N, \tag{6}$$

where N is the population size. Each candidate is evaluated by closed-loop rollouts in the environment, and assigned the score

$$J(w) = 1000 \cdot \text{SuccessRate}(w) + \text{AverageReturn}(w). \tag{7}$$

Let $\mathcal{E}^{(i)}$ denote the set of elite candidates with the highest scores. The search distribution is then updated as

$$\mu^{(i+1)} = \frac{1}{|\mathcal{E}^{(i)}|} \sum_{w \in \mathcal{E}^{(i)}} w, \qquad \sigma^{(i+1)} = \text{Std}\Big(\mathcal{E}^{(i)}\Big) + 10^{-3}, \tag{8}$$

and the best final parameter vector is used as the residual policy for evaluation. Residual finetuning adds approximately 2 hours of training per policy.

4. Results

We evaluate each stage of the pipeline in turn: visual feature representations, labeling model accuracy, the distribution shift challenge, synthetic dataset quality, and downstream policy performance.

4.1. Visual Feature Ablation

We compare CLS token and spatial patch token representations for both the IDM and PEM. Table 1 reports validation MSE on held-out random rollouts.

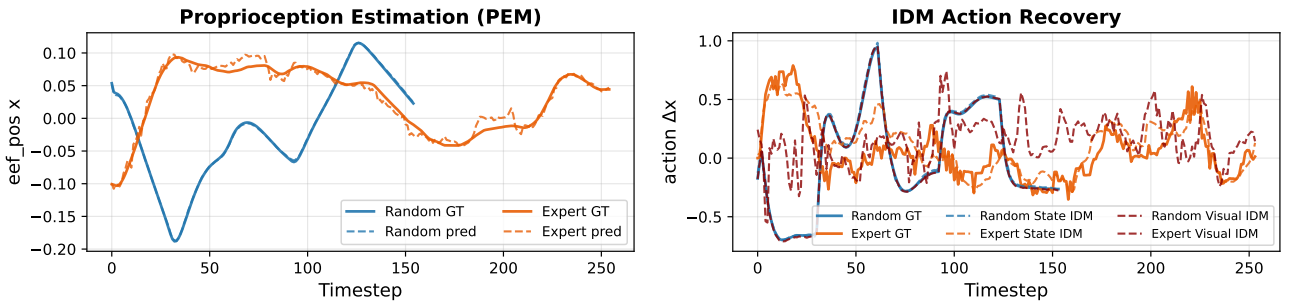


Figure 2. PEM and IDM predictions on held-out random (blue) and expert (orange) trajectories. Solid lines show ground truth; dashed lines show model predictions. **Left:** The PEM accurately tracks proprioception for both distributions. **Right:** The State IDM (dashed) closely recovers ground-truth actions, while the Visual IDM (dotted, dark red) degrades severely on expert trajectories due to distribution shift.

Table 1. Visual feature ablation: validation MSE for PEM and IDM architectures. Spatial patch tokens processed by a transformer dramatically outperform CLS-based MLPs.

Model	Input	Val MSE
Proprioception Estimation Model		
MLP	CLS@480px	4.38×10^{-3}
Transformer	spatial@112px	$\mathbf{5.3 \times 10^{-4}}$
Inverse Dynamics Model		
MLP	CLS@480px	2.81×10^{-4}
MLP	CLS@112px	1.51×10^{-4}
MLP	CLS+spatial@112px	1.15×10^{-4}
Transformer	spatial@112px	$\mathbf{2.2 \times 10^{-5}}$

Spatial patch tokens yield a 5.2× improvement for the IDM and an 8.3× improvement for the PEM over CLS-based MLPs. Notably, the transformer at *lower* resolution (112px) outperforms the MLP at higher resolution (480px), demonstrating that spatial attention over patch tokens is more valuable than raw pixel detail for robotic state estimation. Even coarsely averaging the spatial tokens (CLS+spatial mean) provides a 1.3× improvement over CLS alone, confirming that spatial information is the key factor.

4.2. Distribution Shift in Visual IDMs

While the visual transformer IDM achieves the best validation MSE on random rollouts (Table 1), this does not translate to accurate predictions on expert-like trajectories. Table 2 compares IDM variants on both random and expert data.

The visual transformer IDM has a random-rollout RMSE of 0.004 but an expert RMSE of 0.444, which is worse than even the simple MLP. This severe degradation is caused by distribution shift: the visual statistics of random rollouts (robot waving aimlessly) differ substantially from expert demonstrations (purposeful block manipulation), and the visual IDM overfits to the random distribution. The state IDM avoids this problem entirely, achieving the best expert RMSE (0.192) and the lowest closed-loop pixel difference (7.1) when used to drive a simulated rollout. Figure 2 visualizes this effect: the State IDM closely tracks ground-truth actions on both distributions, while the Visual IDM diverges on expert trajectories.

Table 2. IDM comparison across data distributions. The visual transformer achieves the best random-rollout accuracy but generalizes poorly to expert trajectories. The state IDM, which operates in distribution-invariant proprioception space, is the clear winner on expert data.

Model	Random RMSE	Expert RMSE	Expert CL pixel diff
MLP CLS@480	0.11	0.373	16.0
Transformer spatial@112	**0.004**	0.444	-
State IDM	0.012	**0.192**	**7.1**

4.3. Closed-Loop Evaluation

To test whether the two-stage labeling pipeline (PEM → State IDM) extracts physically plausible actions, we run closed-loop simulation rollouts on all 10 expert demonstrations. We reset the simulator to the ground-truth initial state, then at each timestep feed the live simulator proprioception and the PEM-predicted target proprioception into the State IDM to produce an action. On the 8 training demos, the closed-loop rollouts achieve a mean pixel difference of 2.7 against the ground-truth video, with 5 out of 8 successfully stacking all three blocks. On the 2 held-out validation demos, the mean pixel difference is 6.5, with partial stacking (1-2 blocks), reflecting the compounding effect of PEM

estimation errors on unseen trajectories.

4.4. Synthetic Video Quality

We generate 200 synthetic videos using the finetuned Cosmos model, each conditioned on a first frame from $\mathcal{D}_{\text{random}}$ and labeled via the PEM + State IDM pipeline to produce $\mathcal{D}_{\text{aug}}$ (200 demos, 184 steps each).

Perceptually, the generated videos closely resemble real expert demonstrations: the robot arm moves purposefully, grasps and places blocks in sequence, and the visual quality is largely indistinguishable from simulator renders. To a human observer, many videos depict convincing task completions. However, when we execute the extracted actions in closed-loop simulation, the outcomes frequently differ from what the video depicts (Figure 3). A common failure mode is that the arm passes slightly too high above the target block during a grasp attempt. This points to a fundamental limitation: the WFM produces videos that are visually realistic but not necessarily physically grounded, and small spatial inaccuracies in the generated motion compound into qualitatively different outcomes.

Beyond spatial precision, we observe several categories of outright hallucination in the generated videos: blocks ghosting through other blocks or the table surface, duplication of blocks (a third copy appearing mid-sequence), incorrect stacking order relative to the prompt, and blocks levitating without gripper contact. These hallucinations (Figure 4) are relatively infrequent but highlight that the WFM lacks an internal physics model and instead relies on learned visual priors that can be violated.

4.5. Policy Performance

We evaluate downstream policies on `StackThree` using staged success metrics rather than only binary full-task completion. Specifically, we report *Pickup-1*, whether the policy successfully picks up the first cube; *Stack-1*, whether it successfully places the first cube; and *Stack-2*, whether it successfully places the second cube to complete the full three-block stack. This breakdown is useful because synthetic-data-trained policies often learn coarse task structure without achieving reliable long-horizon completion.

To test the utility of synthetic demonstrations, we train Diffusion Policies in `robomimic` under multiple expert / synthetic data mixtures: 200 expert, 200 synthetic, 50 expert + 150 synthetic, 100 expert + 100 synthetic, 10 expert, and 10 expert + 100 synthetic demonstrations. Each policy is trained for 1000 epochs on a single NVIDIA RTX 5050 GPU, requiring approximately 9 hours per run. We evaluate the final policies after residual RL finetuning via CEM, which adds roughly 2 hours of training per policy.

A consistent pattern across experiments is that expert-only training remains the strongest regime, synthetic-only training performs the worst, and mixed-data policies lie in between. Among the mixed settings, 100 expert + 100 synthetic substantially outperforms 50 expert + 150 synthetic, suggesting that synthetic demonstrations cannot fully substitute for high-quality expert supervision even when the total number of demonstrations is held fixed. Synthetic data appears to preserve enough task structure to support some early-stage progress, but these gains degrade sharply on later stages that require precise physical interaction. This staged breakdown supports our broader conclusion that the synthetic dataset preserves coarse task intent better than the fine-grained physical accuracy required for reliable three-block stacking.

Table 3. Final policy performance on `StackThree` after behavioral cloning followed by residual RL finetuning, under different training data mixtures.

Training data	Pickup-1 (%)	Stack-1 (%)	Stack-2 (%)
200 expert	80	75	60
200 synthetic	0	0	0
50 expert + 150 synthetic	20	20	0
100 expert + 100 synthetic	45	35	20
10 expert	20	0	0
10 expert + 100 synthetic	0	0	0

The final BC+residual-RL policies show that residual finetuning is not sufficient to overcome systematic errors in the synthetic demonstrations. When the policy is initialized from datasets with substantial synthetic content, performance remains far below the expert-only setting, especially on the later stages of the task. This suggests that while residual finetuning may improve local robustness, it cannot fully recover from a weak behavioral-cloning initialization induced by physically misaligned synthetic trajectories.

5. Discussion

Physical groundedness is the central challenge. Our results reveal that the primary bottleneck in using WFMs as data engines is not perceptual quality (the finetuned Cosmos model produces visually impressive and temporally coherent videos) but rather *physical groundedness*. The generated videos are not constrained by physics: the robot arm may pass slightly above a block yet the video depicts a successful grasp, blocks ghost through surfaces, and objects duplicate or vanish. These physically implausible trajectories, when labeled and used as policy data, introduce systematic errors that no downstream labeling model can fully correct.

Two-stage labeling is necessary but bounded by video quality. Our PEM + State IDM architecture addresses

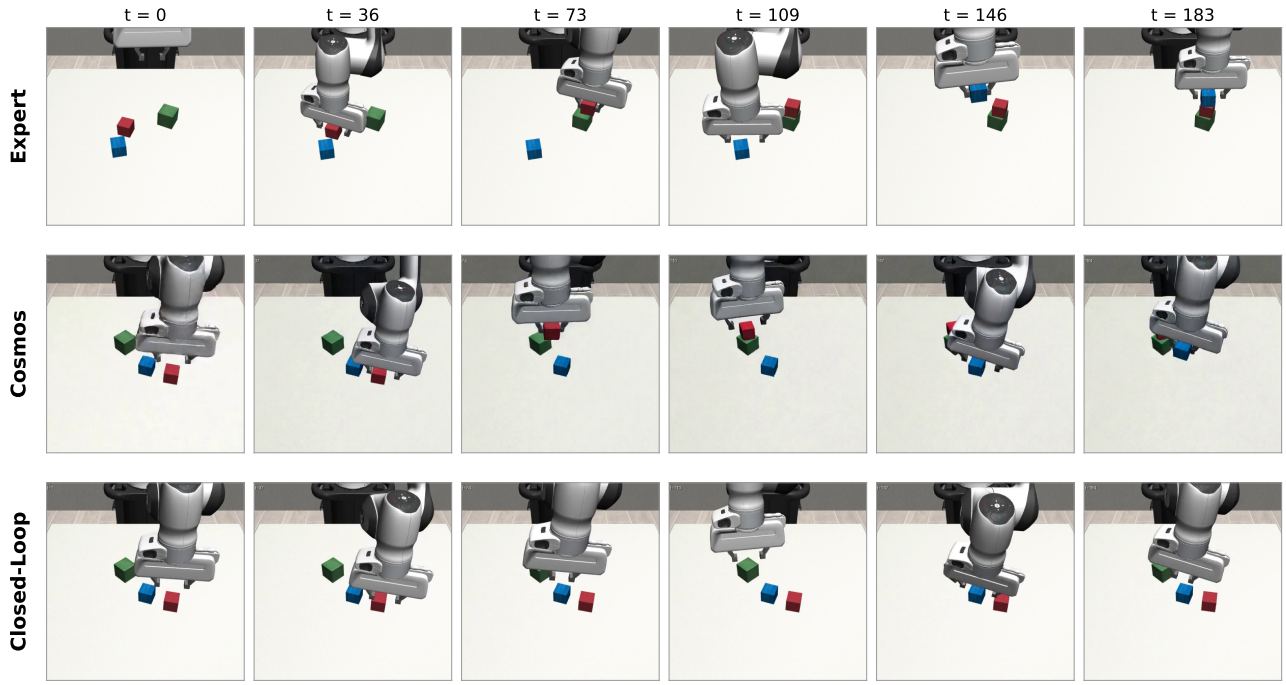
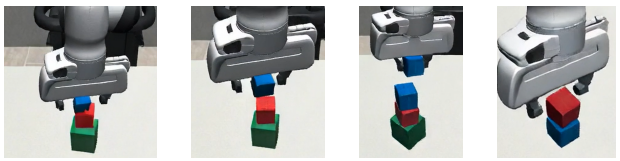


Figure 3. Comparison of a real expert demonstration (top), a Cosmos-generated synthetic video (middle), and the closed-loop simulator rollout driven by the extracted actions (bottom). The Cosmos video depicts visually plausible block manipulation, but the closed-loop rollout reveals physical misalignment: the gripper fails to make contact during grasp attempts, resulting in a different outcome.



(a) Ghosting (b) Levitation (c) Duplication (d) Wrong order

Figure 4. Examples of WFM hallucinations in Cosmos-generated videos. Despite producing visually realistic scenes, the model violates physical constraints: blocks clip through each other (a), blocks levitate without support (b), extra blocks appear mid-sequence (c), and blocks are stacked in the wrong order (d).

the distribution shift that arises in labeling, reducing expert RMSE from 0.444 (visual IDM) to 0.192 (state IDM), a 2.3× improvement. This two-stage decomposition bridges the gap between pixels and actions. However, the labeling pipeline is fundamentally bounded by the quality of the upstream video: no labeling model can recover physically valid actions from a video that depicts physically impossible events.

Closed-loop evaluation as a diagnostic tool. Our methodology of comparing Cosmos-generated video against closed-loop simulator rollouts driven by extracted actions (Section 4, Figure 3) provides a systematic way to expose the gap between what a WFM "imagines" and what actually happens under physics. This evaluation protocol may be useful more broadly for assessing the physical fidelity of video generation models intended for robotics applications.

Spatial representations matter. The consistent 5-8× improvement from spatial patch tokens over CLS tokens across both IDM and PEM tasks suggests that fine-grained spatial information is critical for robotic state estimation from images. This finding has implications beyond our pipeline: any method that extracts robot or object state from pretrained visual features should prefer spatial patch tokens over global summaries.

Synthetic data helps coarse progress more than full completion. Our downstream policy experiments show that synthetic demonstrations can encode enough structure to support early-stage task progress, especially when measured with staged success metrics. However, these gains do not reliably translate to full-task completion. Policies trained on synthetic or mixed data may learn the rough sequence of motions required for stacking, yet still fail at the precise geometric alignment and contact timing needed for successful long-horizon manipulation. This reinforces the central

theme of the paper: the limiting factor is not visual realism alone, but whether the generated trajectories are physically accurate enough to serve as supervision for control.

Future directions. The core limitation of our approach is that video-only WFMs lack physical grounding. Recent work has begun to address this by jointly modeling video and actions within a single generative process. Cosmos Policy (Kim et al., 2026) and DreamZero (Ye et al., 2026) both jointly model video and actions within a single diffusion process, making actions a first-class output of the generative model and eliminating the need for post-hoc action labeling entirely. Other promising directions include multi-view video generation to provide geometric constraints that reduce spatial ambiguity, and incorporating explicit physics priors or simulator-in-the-loop verification during generation.

6. Conclusion

We presented a pipeline for generating labeled synthetic demonstrations from a world foundation model, requiring only 10 expert demonstrations and 1000 cheaply collected random rollouts. Our key finding is that while WFMs produce visually realistic videos, the lack of physical groundedness, not perceptual quality, is the primary barrier to generating effective policy data. Subtle spatial inaccuracies and outright physical hallucinations in the generated videos introduce systematic errors that propagate through any downstream labeling pipeline. To mitigate the labeling challenges that arise in this setting, we proposed a two-stage architecture (PEM + State IDM) that operates in distribution-invariant proprioception space, reducing expert-trajectory action RMSE by 2.3$\times$ compared to visual inverse dynamics. Our ablation studies additionally demonstrate that spatial patch tokens from pretrained vision encoders yield 5-8$\times$ improvements over CLS tokens for robotic state estimation. We believe that jointly modeling actions alongside video, as in Cosmos Policy and DreamZero, is the most promising path toward closing the physical grounding gap and making WFM-generated data truly useful for policy learning.

7. Contributions

Albert Lin: Implemented the synthetic dataset generation pipeline, including finetuning the Cosmos Predict 2 WFM on Google Cloud, training the PEM and IDM with multiple architectures, and building the end-to-end labeling pipeline. Conducted the visual feature ablation study comparing CLS and spatial patch token representations. Evaluated synthetic data quality via open-loop and closed-loop simulation.

Aarya Sumuk: Implemented the downstream policy training pipeline, including Diffusion Policy training in `robomimic` across expert-only, synthetic-only, and mixed-data regimes on the `StackThree` task. Implemented residual RL finetuning via a CEM-based residual MLP on top of a frozen BC-trained diffusion policy. Conducted the policy evaluation study using rollout experiments and staged success metrics.

8. Acknowledging the Use of AI Tools

We are responsible for all the content contained in this report. However, we would like to acknowledge the use of Claude Opus 4.6 for help with the preparation of this final report, as well as for the generation of code used in this project.

References

Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11): 1684–1704, 2025.

Du, Y., Yang, M., Dai, B., Dai, H., Nachum, O., Tenenbaum, J. B., Schuurmans, D., and Abbeel, P. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse control tasks through world models. *Nature*, 640: 647–653, 2025.

Jang, J., Ye, S., Lin, Z., Xiang, J., Bjorck, J., Fang, Y., Hu, F., Huang, S., Kundalia, K., Lin, Y.-C., et al. DreamGen: Unlocking generalization in robot learning through video world models. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5170–5194. PMLR, 2025.

Kim, M. J., Gao, Y., Lin, T.-Y., Lin, Y.-C., Ge, Y., Lam, G., Liang, P., Song, S., Liu, M.-Y., Finn, C., and Gu, J. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., and

Martín-Martín, R. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

Mandlekar, A., Nasiriany, S., Wen, B., Akinola, I., Narang, Y., Fan, L., Zhu, Y., and Fox, D. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.

NVIDIA, Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., et al. Cosmos world foundation model platform for physical AI, 2025.

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al. DINOv3, 2025.

Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Kaelbling, L., Schuurmans, D., and Abbeel, P. UniSim: Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024.

Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Zhou, G., Pan, H., LeCun, Y., and Pinto, L. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

Zhu, Y., Wong, J., Mandlekar, A., Martín-Martín, R., Joshi, A., Nasiriany, S., Zhu, Y., and Lin, K. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.