Next Forcing: World Action Modeling with Multi-Chunk Prediction

Gangwei Xu[1,2] **Qihang Zhang**[1,†] **Jiaming Zhou**[1,4] **Xing Zhu**[1]
Yujun Shen[1] **Xin Yang**[2,5,‡] **Yinghao Xu**[3,1,‡]

[1]Robbyant [2]HUST [3]HKUST [4]HKUST (GZ) [5]Optics Valley Laboratory

[†]Project Lead [‡]Corresponding Author

Abstract

Autoregressive video generation has emerged as a powerful paradigm for World Action Models (WAMs). However, existing approaches suffer from slow training convergence and limited converged accuracy, particularly at high frame rates, as the training supervision is confined to the current chunk without explicit signals about future dynamics; they also suffer from slow inference due to iterative video denoising. In this paper, we present Next Forcing, a multi-chunk prediction (MCP) framework for world action modeling that enables faster training, higher accuracy, and accelerated inference. Inspired by multi-token prediction in large language models, Next Forcing introduces an MCP training objective that augments the main model with lightweight auxiliary MCP modules to simultaneously denoise video chunks at multiple future temporal horizons (next^1, next^2, next^3 chunks). These MCP modules form a causal chain across prediction depths, where intermediate features fused from multiple layers of the main model are leveraged to predict future dynamics, allowing near-future predictions to inform farther-future ones and providing dense multi-scale temporal supervision back to the main model. During training, the MCP modules significantly accelerate convergence and improve converged accuracy, especially at high frame rates: at 50 fps, Next Forcing achieves a 93.1% relative improvement over LingBot-VA at 5k training steps and 2.3$\times$ faster convergence, and establishes new state-of-the-art results on the RoboTwin benchmark (94.1/93.5% on Clean/Random). At inference, the MCP modules can be retained to predict the next video chunk in parallel with the current one, achieving 2$\times$ inference acceleration. Beyond World Action Models, Next Forcing also improves physical-law adherence on PhyWorld and reduces FVD by over 50% in action-free general video pretraining. Project website: `https://gangweix.github.io/next-forcing/`.

1 Introduction

Videos capture how the physical world evolves and how agents act within it, recording the dynamics of physical interactions at scale [24, 25, 26, 51, 9, 1, 2]. Building on this, World Action Models (WAMs) [36, 63] have recently emerged as a new paradigm for embodied AI, learning manipulation policies by jointly modeling future video and actions. The dominant training objective for WAMs is *teacher-forced next-chunk denoising*, where the model denoises the noisy current chunk conditioned on clean past chunks from ground-truth data. Despite the recent achievements of this paradigm [36, 63], teacher-forced next-chunk denoising remains an inefficient and shortcut-prone training signal for video world models. More precisely, predicting the next chunk is a fundamentally *local* task, which admits an appearance shortcut [21]: since adjacent chunks are visually highly similar, much of the denoising loss can be driven down by learning a near-identity map from the clean past

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

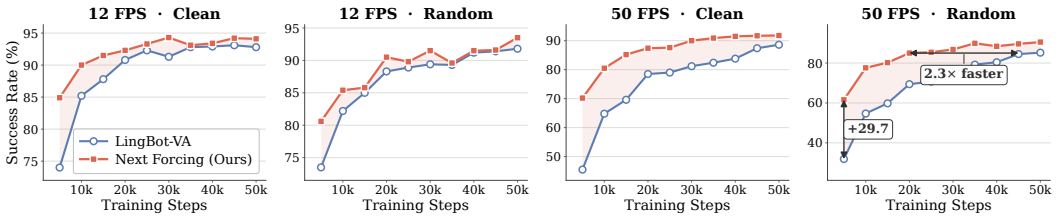


Figure 1: **Task success rate (%) on RoboTwin across training steps.** Next Forcing converges faster and reaches higher final accuracy than LingBot-VA at both 12 and 50 fps. The advantage is most pronounced at 50 fps: at 5k steps Next Forcing already outperforms LingBot-VA by 29.7 points on Random, and matches its 45k-step accuracy at only 20k steps, a 2.3× training speedup.

chunk to the current one, with only small residual corrections. Such a shortcut is much easier to fit than the underlying dynamics and can absorb a large share of the gradient signal, leaving weak pressure on the model to capture the long-range temporal evolution that governs how the scene actually changes. We refer to this as *myopic supervision*, and its cost becomes particularly acute at high frame rates: at 50 fps, the appearance gap between adjacent chunks narrows to the point where the shortcut is nearly lossless, and standard teacher forcing converges significantly slower with lower final accuracy (Figure 1).

Our central insight is that turning the local single-chunk objective into a *long-range multi-chunk* objective forces the model to learn the latent dynamics governing temporal evolution rather than relying on appearance shortcuts. This idea has been validated in language modeling, where multi-token prediction (MTP) [18, 23] trains auxiliary modules to predict multiple future tokens, improving sample efficiency and enabling inference acceleration. Adapting it to video world models, however, is non-trivial: prediction targets are *continuous* video latents rather than discrete language tokens, generation proceeds through *iterative denoising* rather than single-step sampling, and temporal dependencies span *multiple horizons* of varying scale.

We present **Next Forcing**, a multi-chunk prediction (MCP) framework that addresses myopic supervision for better and faster world action modeling. We use *world action modeling* to refer to jointly modeling future visual dynamics and robot actions within a unified generative architecture. Next Forcing augments the main model with a small set of auxiliary MCP modules that simultaneously predict video chunks at multiple future horizons (next^1, next^2, next^3) and form a causal chain across prediction depths. To let the temporal supervision propagate deep into the main model's representations, the MCP modules fuse intermediate features from *multiple layers* of the main model rather than the final layer alone. The MCP modules are trained with a *higher timestep shift* than the main model, forcing them to rely more heavily on the main model's representations and tightening the coupling between the two. Although the MCP objective directly supervises the video stream, the improved future representations propagate to the action stream through cross-modal attention.

Our work joins a growing family of "forcing" methods for autoregressive video generation. Existing methods vary either *what context the model sees* (as in teacher forcing [36, 63] and self forcing [28]) or *how noise is scheduled* (as in diffusion forcing [13]). Next Forcing is orthogonal to both: it varies *what the model is asked to predict*, and is therefore composable with all of the above.

We primarily evaluate Next Forcing as a World Action Model on RoboTwin [15]. We further evaluate its action-free form on PhyWorld [31] and general video pretraining to verify that its temporal supervision benefits extend beyond robot action learning. Next Forcing sets new state-of-the-art on RoboTwin (94.1/93.5% on Clean/Random), achieves 2.3× faster convergence at 50 fps (Figure 1), 2× inference acceleration, substantial gains on PhyWorld, and over 50% FVD reduction on general video pretraining with 3.5M in-house video clips, confirming its effectiveness beyond robot-specific data. Our main contributions are as follows:

- We propose **Next Forcing**, a multi-chunk prediction framework for World Action Models that addresses the myopic supervision of autoregressive video prediction.
- Next Forcing establishes new state-of-the-art results on the RoboTwin benchmark with significantly faster training convergence. At 50 fps, Next Forcing achieves 2.3× faster convergence and higher converged accuracy.

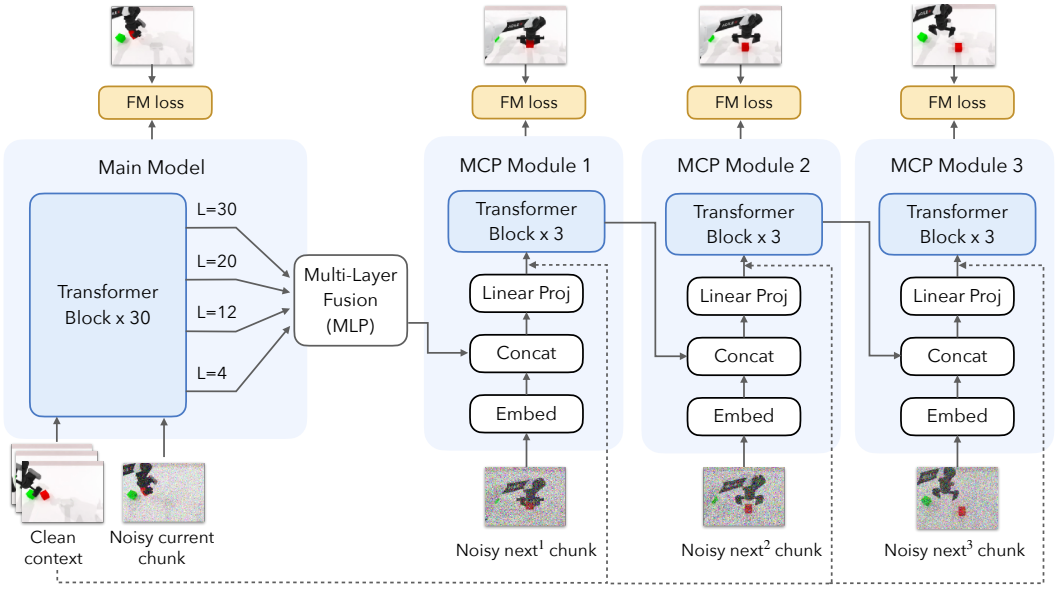


Figure 2: **Overview of Next Forcing.** The main model denoises the current chunk, while chained MCP modules predict future chunks (next[1], next[2], . . .) using features from the main model, providing dense temporal supervision during training and enabling parallel chunk prediction at inference.

- Beyond improving training, the MCP modules can be retained at inference to predict the next video chunk in parallel with the current one, accelerating generation.
- We provide systematic ablations on the design choices that enable multi-chunk prediction to transfer from discrete tokens to continuous video latents (timestep shift, multi-layer feature fusion, and causal MCP chain), offering practical guidance for future work (Section 5).

We hope Next Forcing motivates further investigation of training *objectives* (beyond context construction and noise scheduling) in autoregressive video generation.

2 Related Work

2.1 World Action Models

Unlike Vision-Language-Action (VLA) models [8, 33, 71, 7, 47, 46, 22, 42, 43, 17, 69, 41, 72, 68, 48, 59, 44, 62, 58, 73] that directly map visual observations and language instructions to actions, World Action Models (WAMs) [63, 36, 29, 53, 11, 61, 12, 70, 3] incorporate video prediction into robot policy learning, first predicting future visual dynamics and then decoding robot actions from the predicted frames. Several recent works have advanced the WAM paradigm along different axes. DreamZero [63] demonstrates that a 14B-parameter autoregressive video diffusion model, trained on diverse non-repetitive robot data, achieves zero-shot generalization to novel tasks and cross-embodiment transfer. LingBot-VA [36] proposes a unified autoregressive framework with teacher forcing that jointly learns video prediction and action execution, achieving state-of-the-art bimanual manipulation through closed-loop control with persistent KV cache memory. Concurrently, Fast-WAM [66] finds that the primary benefit of video modeling lies in improving world representations during training. Other works explore joint video-action generation from complementary perspectives, including latent action representations[6, 16, 64, 52], joint video-action diffusion [74, 60], video-conditioned policy learning [19, 27, 5, 37, 55, 34], and video generation as policy [32, 45]. Our work is complementary to these efforts: rather than proposing a new WAM architecture or exploring data scaling, we focus on improving the *training paradigm* of autoregressive WAMs through multi-chunk prediction, which is applicable to existing WAM frameworks and also enables inference acceleration.

2.2 Autoregressive Video Generation

Autoregressive video generation [30, 65, 40, 14, 50] has become the core generation paradigm for action-conditioned world models and world action models [36, 63, 32, 67]. The standard training approach is teacher forcing [36, 63], where the model learns to denoise the current chunk conditioned on clean ground-truth context. While stable, it creates a distribution gap between training (clean context) and inference (self-generated context), known as exposure bias [49, 4, 28]. Diffusion Forcing [13] mitigates this by training with independently sampled noise levels per frame, so that the model encounters noisy context during training. Self Forcing [28] takes a more direct approach by conditioning on self-generated histories during training, explicitly bridging the train-test distribution gap through a distribution-matching loss. These methods primarily address *how context is constructed* during training or inference. In contrast, our Next Forcing addresses a different and complementary aspect: *what the model is trained to predict*. By extending the prediction target from the current chunk to multiple next chunks, Next Forcing provides dense multi-scale temporal supervision that encourages trajectory-level temporal reasoning and avoids myopic supervision.

3 Preliminaries

3.1 Flow Matching

Flow matching [39] is a generative modeling framework that learns a velocity field to transport samples from a noise distribution to the data distribution. Given a clean sample $\mathbf{x}_0$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, the noisy sample at time $t \in [0, 1]$ is constructed via linear interpolation:

$$\mathbf{x}_t = (1 - t)\,\mathbf{x}_0 + t\,\boldsymbol{\epsilon}. \tag{1}$$

A neural network $v_\theta(\mathbf{x}_t, t, \mathbf{c})$ is trained to predict the velocity $\mathbf{v}^* = \boldsymbol{\epsilon} - \mathbf{x}_0$ conditioned on context $\mathbf{c}$, with the training objective:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\left\| v_\theta(\mathbf{x}_t, t, \mathbf{c}) - (\boldsymbol{\epsilon} - \mathbf{x}_0) \right\|^2 \right]. \tag{2}$$

The timestep t is sampled with a timestep shift parameter s (see Appendix C for the detailed formulation). At inference, clean samples are generated by integrating the velocity field from $t = 1$ (pure noise) to $t = 0$ (clean data) using an ODE solver.

3.2 Autoregressive Video Generation with Teacher Forcing

We build upon the autoregressive video-action framework of LingBot-VA [36]. The model operates on video latents encoded by a pre-trained VAE and generates video frames in chunks of M frames. At each autoregressive step i, the model denoises the noisy current chunk $\mathbf{x}_t^{(i)}$ conditioned on the clean previous chunks $\mathbf{x}_0^{(1:i-1)}$ from ground-truth data and a language instruction ℓ:

$$v_\theta\left(\mathbf{x}_t^{(i)}, t, \left[\mathbf{x}_0^{(1:i-1)}, \ell\right]\right) \approx \boldsymbol{\epsilon}^{(i)} - \mathbf{x}_0^{(i)}. \tag{3}$$

This teacher forcing formulation naturally aligns with closed-loop deployment, where ground-truth observations replace generated frames after each action execution. However, as discussed in Section 1, the model's supervision is confined to the current chunk, leading to the myopic supervision problem.

4 Next Forcing

4.1 Overview

Next Forcing augments the standard teacher-forcing training objective with multi-chunk prediction (MCP), where auxiliary modules predict video chunks at multiple next temporal horizons in addition to the current chunk. Figure 2 illustrates the overall framework. During training, the main model denoises the current chunk as in standard teacher forcing, while three auxiliary MCP modules denoise the next^1, next^2, and next^3 chunks through a causal chain, where each depth builds on the output of the previous one. At inference time, the MCP modules can either be discarded for zero-overhead deployment or retained for parallel chunk generation. Our primary instantiation is a World Action Model that jointly predicts future visual dynamics and robot actions.

4.2 Multi-Chunk Prediction Objective

The key idea of Next Forcing is to extend the prediction target from the current chunk to multiple next chunks, providing explicit supervision about future dynamics. This encourages trajectory-level temporal reasoning during training and directly addresses the myopic supervision problem.

Temporal Chunk Shifting. Given a training video latent $\mathbf{x}_0 \in \mathbb{R}^{C\times F\times H\times W}$, where C, F, H, W denote the channel, number of chunks, height, and width dimensions respectively. Each chunk contains M frames, where M is randomly sampled from $\{1, \ldots, M_{\text{max}}\}$ at each training step for robustness across temporal scales. Let i denote the current chunk index. For each MCP depth $k \in \{1, 2, 3\}$, we construct a temporally shifted target $\mathbf{x}_0^{[k]}$ by advancing the video latent by k chunks:

$$\mathbf{x}_0^{[k]}[i] = \mathbf{x}_0[\min(i + k,\, F)]\,, \tag{4}$$

where chunks beyond the sequence boundary are padded by replicating the last chunk. Each $\mathbf{x}_0^{[k]}$ is the video shifted k chunks into the future.

Independent Noise Injection. Each shifted target is independently noised using the flow matching formulation with its own timestep and noise sample:

$$\mathbf{x}_{t_k}^{[k]} = (1 - t_k)\,\mathbf{x}_0^{[k]} + t_k\,\boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \mathbf{I}), \tag{5}$$

where t_k is sampled with a dedicated timestep shift parameter s_{mcp}. We set $s_{\text{mcp}} > s_{\text{main}}$, which biases the MCP modules toward higher noise levels. The motivation is to strengthen the coupling between the MCP modules and the main model: at higher noise levels the MCP input carries less information about its own target, so the modules are forced to rely more heavily on the main model's representations to denoise. This pushes the MCP loss gradients into the main model rather than letting the lightweight MCP modules absorb the supervision themselves.

MCP Position Encoding. To inform the MCP modules of their temporal offset, the chunk shift is incorporated into the rotary position embeddings (RoPE) [54]:

$$\text{RoPE}(\mathbf{x}_0^{[k]}[i]) = \text{RoPE}(i + k), \tag{6}$$

so that each MCP module receives positional information for the future chunk it predicts.

4.3 Chained MCP Modules

The MCP modules are designed to encourage the main model to develop trajectory-level temporal representations. To achieve this, we fuse intermediate features from multiple layers of the main model rather than only the final output, since early layers tend to capture coarse structural patterns while later layers refine fine-grained details. This multi-layer fusion allows the MCP supervision signal to influence the main model's representations at various depths through gradient backpropagation.

Multi-Layer Feature Fusion. During the main model's forward pass through its 30 transformer layers, we collect hidden states at 4 intermediate layers $\{4, 12, 20, 30\}$, capturing representations at various depths of the backbone. Notably, the collected hidden states include both the noisy current latent and the clean history latent, allowing the fusion to capture both the denoising state and the ground-truth context. These multi-scale features are concatenated along the feature dimension and compressed through a two-layer MLP:

$$\mathbf{h}_{\text{fuse}} = \text{MLP}([\mathbf{h}_4; \mathbf{h}_{12}; \mathbf{h}_{20}; \mathbf{h}_{30}]) \in \mathbb{R}^{B\times N\times d}, \tag{7}$$

where N is the number of latent tokens and d is the hidden dimension. During backpropagation, the MCP loss gradients flow through $\mathbf{h}_{\text{fuse}}$ back into these intermediate layers, providing temporally-aware supervision to both the early and late stages of the main model.

Causal Chain Across Depths. The three MCP modules form a causal chain. For each depth k, the noisy shifted target $\mathbf{x}_{t_k}^{[k]}$ is embedded through the shared patch embedding layer and fused with the output from the previous depth:

$$\mathbf{z}^{[k]} = W_k \left[\mathbf{h}_{\text{prev}}^{[k-1]};\, \text{Embed}(\mathbf{x}_{t_k}^{[k]})\right], \quad W_k \in \mathbb{R}^{d\times 2d}, \tag{8}$$

where $\mathbf{h}_{\text{prev}}^{[0]} = \mathbf{h}_{\text{fuse}}$. The fused representation is processed through 3 lightweight transformer blocks to predict the flow matching velocity $\hat{\mathbf{v}}^{[k]}$. The output simultaneously serves as $\mathbf{h}_{\text{prev}}^{[k]}$ for the next depth, allowing depth-2 predictions to build on depth-1 features, and depth-3 on depth-2. The MCP modules share the same attention mask as the main model, requiring only a single mask construction per training step for efficient training (see Appendix A for details).

4.4 Joint Video-Action Architecture

Following LingBot-VA [36], Next Forcing jointly models video prediction and action decoding. The joint prediction is decomposed into two stages: (1) predicting future visual dynamics, and (2) decoding actions via inverse dynamics:

$$\mathbf{x}_{i+1} \sim p_\theta(\cdot \mid \mathbf{x}_{\leq i}, \mathbf{a}_{<i}, \ell), \quad \mathbf{a}_i \sim g_\psi(\cdot \mid \mathbf{x}_{\leq i+1}, \mathbf{a}_{<i}, \ell), \tag{9}$$

where $\mathbf{x}_{\leq i}$ denotes the observation history up to chunk i, $\mathbf{a}_{<i}$ denotes the action history, and ℓ is the language instruction. The video stream first predicts the next visual state $\mathbf{x}_{i+1}$, then the action stream decodes $\mathbf{a}_i$ by conditioning on the observations including the predicted future chunk $\mathbf{x}_{i+1}$.

Both streams are implemented within a unified Mixture-of-Transformers (MoT) [38] architecture, where a video stream and an action stream fuse through cross-modal attention at each transformer layer. The MCP modules introduced in Section 4.3 operate on the video stream. The improved video representations propagate to the action stream through the shared cross-modal attention, benefiting action decoding indirectly. Therefore, although the MCP losses are directly applied to the video stream, their temporally enriched representations improve joint world-action learning and action decoding through cross-modal interaction.

4.5 Training Objective

Main Loss. The main model is trained with two flow matching losses for the video and action streams. The video dynamics loss supervises the video stream to predict future visual states:

$$\mathcal{L}_{\text{video}} = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}} \left[\left\| v_\theta(\mathbf{x}_t, t, \mathbf{c}) - (\boldsymbol{\epsilon} - \mathbf{x}_0) \right\|^2 \right], \tag{10}$$

and the action loss supervises the action stream via inverse dynamics:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{t,\mathbf{a}_0,\boldsymbol{\epsilon}} \left[\left\| v_\psi(\mathbf{a}_t, t, \mathbf{c}_a) - (\boldsymbol{\epsilon} - \mathbf{a}_0) \right\|^2 \right], \tag{11}$$

where $\mathbf{a}_0$ denotes the ground-truth action sequence and $\mathbf{c}_a$ includes the visual context from both current and future observations.

MCP Loss. Each MCP module is supervised with a flow matching loss on its temporally shifted target $\mathbf{x}_0^{[k]}$ (Eq. 4). For depth k:

$$\mathcal{L}_k^{\text{MCP}} = \mathbb{E}_{t_k,\mathbf{x}_0^{[k]},\boldsymbol{\epsilon}_k} \left[\left\| v_\theta^{[k]}(\mathbf{x}_{t_k}^{[k]}, t_k, \mathbf{c}) - (\boldsymbol{\epsilon}_k - \mathbf{x}_0^{[k]}) \right\|^2 \right], \tag{12}$$

where the last k padded chunks are excluded from the loss computation.

Total Loss. The complete training objective combines the main losses and the MCP losses:

$$\mathcal{L} = \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}} + \sum_{k=1}^{3} w_k \cdot \mathcal{L}_k^{\text{MCP}}. \tag{13}$$

4.6 Inference

The MCP modules introduced in Section 4.3 serve a dual role: they are trained as a supervision mechanism but can also be repurposed at inference. We support two inference modes that share the same trained checkpoint and trade off cost against throughput.

Table 1: Evaluation on the RoboTwin benchmark (average success rate % over 50 tasks). Clean: fixed initial configurations. Random: randomized object poses and scene layouts. Best results are **bolded**.

	X-VLA	π_0	$\pi_{0.5}$	Motus	Being-H0.7	Fast-WAM	LingBot-VA	**Next Forcing**
Clean	72.9	65.9	82.7	88.7	90.2	91.9	92.9	**94.1**
Random	72.8	58.4	76.8	87.0	89.6	91.8	91.5	**93.5**

Table 2: Ablation studies on RoboTwin Clean subset, trained for 20k steps on 16 GPUs. Default baseline: s_{main}=5 with noisy history augmentation. Default MCP: s_{mcp}=10, multi-layer fusion, weight init from main model, 3 transformer blocks per depth.

Baseline (LingBot-VA [36]) Ablation		MCP Module Ablation	
Configuration	SR (%)	Configuration	SR (%)
Baseline (default)	75.6	Baseline + MCP (default)	85.8
s_{main}=1	65.3	s_{mcp}=5	83.2
s_{main}=10	78.4	w/o multi-layer fusion	83.6
s_{main}=20	77.6	w/o weight init	83.8
s_{main}=25	77.2	transformer blocks=1	86.5
w/o noisy history aug.	69.8	transformer blocks=5	85.0

Zero-Overhead Mode. All MCP modules (the fusion MLP, projection layers, and lightweight transformer blocks) are discarded. The main model operates exactly as in the standard autoregressive pipeline: at each step, it denoises the current chunk, with the same architecture, latency, and memory footprint as the baseline. All quality gains in this mode come from the enriched training signal that the MCP objective propagates back into the main model during training, with no test-time cost.

Parallel Chunk Generation Mode. The MCP modules can also be retained at inference to amortize the cost of one main-model forward pass over multiple chunks, in spirit similar to speculative and parallel decoding for LLMs [35, 10]. Concretely, in a single denoising trajectory the main model produces the current chunk while the depth-1 MCP module simultaneously produces the next chunk, and the MCP transformer blocks are an order of magnitude lighter than the main model, so adding them to the forward pass is nearly free. Each autoregressive step therefore advances the video by two chunks instead of one, yielding $2\times$ inference acceleration. The depth-2 and depth-3 MCP modules are not used in this mode, as their predictions are superseded by the main model in the next step, but the same mechanism could be extended for higher speedups at the cost of accumulated drift; we leave this exploration to future work.

The two modes use the *same* trained model, allowing deployments to choose freely between strict baseline parity and $2\times$ throughput without retraining.

5 Experiments

5.1 Experimental Setup

Benchmarks. We evaluate Next Forcing on two benchmarks. **RoboTwin** [15] is a simulation benchmark containing 50 bimanual manipulation tasks that require coordinated dual-arm control. Each task is evaluated under two settings: *Clean* (fixed initial configurations) and *Random* (randomized object poses and scene layouts). We report the average task success rate over all 50 tasks. RoboTwin serves as our primary evaluation of world action modeling, where we measure closed-loop manipulation performance using task success rate. **PhyWorld** [31] evaluates whether video generation models can discover and adhere to physical laws (e.g., uniform linear motion, elastic collision, parabolic motion) from visual data alone.

Implementation Details. Next Forcing is built on the LingBot-VA framework with a Wan2.2 [57] Transformer backbone (30 layers). We keep all baseline settings identical to LingBot-VA. The main model uses a timestep shift of $s_{\text{main}} = 5$ with noisy history augmentation applied at probability 0.5. For the MCP modules, we use 3 prediction depths (next^1, next^2, next^3), each with 3 transformer

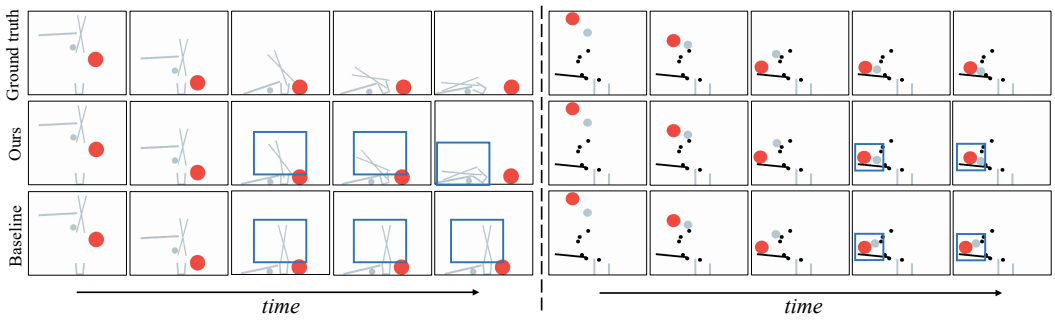


Figure 3: **Qualitative comparison on PhyWorld.** We show 5 frames (start, 3 intermediate, end) from ground truth (top), Next Forcing (middle), and Baseline (bottom). Blue boxes highlight regions where the baseline deviates from the ground-truth physical trajectory, while Next Forcing generates more physically consistent dynamics.

blocks. The MCP timestep shift is set to $s_{\text{mcp}} = 10$. MCP loss weights are $w_1 = 0.5$, $w_2 = 0.2$, $w_3 = 0.1$. The chunk size M is randomly sampled from $\{1, \ldots, M_{\text{max}}\}$ with $M_{\text{max}} = 4$. MCP module weights are initialized from the last few layers of the main model. Following LingBot-VA, we first pre-train on a large-scale multi-embodiment dataset and then post-train on RoboTwin. We adopt the same training setup as LingBot-VA [36], training on 2,500 Clean demonstrations (50 per task) combined with 25,000 Random demonstrations (500 per task) for up to 50k steps. We train on 64 GPUs and evaluate at multiple frame rates to study the effect of frame rate on training convergence and inference acceleration. For ablation studies, we train on 16 GPUs using only the 2,500 Clean demonstrations at 25 fps for 20k steps for efficient comparison.

5.2 Main Results

5.2.1 Comparison with State-of-the-Art

Table 1 compares Next Forcing with state-of-the-art methods on the RoboTwin benchmark. Next Forcing achieves the highest success rate on both Clean (94.1%) and Random (93.5%) settings, outperforming all compared methods including both VLA and WAM approaches.

5.2.2 Training Convergence Analysis

Figure 1 presents a detailed comparison of training convergence between Next Forcing and LingBot-VA across two frame rates and two difficulty levels (full numerical results in Appendix B).

At 12 fps, Next Forcing converges roughly 2× faster than LingBot-VA, reaching 90.0% at 10k steps versus the baseline's ∼20k steps, and achieves higher final accuracy (94.1/93.5% vs. 92.8/91.8% on Clean/Random). The advantage becomes dramatically more pronounced at 50 fps: at 5k steps, Next Forcing already achieves 70.2/61.6% versus LingBot-VA's 45.5/31.9%, and maintains the lead through convergence. At 50k steps, Next Forcing reaches 91.8/90.5% versus 88.6/85.2%. The large gap at high frame rates confirms that multi-chunk prediction is particularly effective when adjacent frames are nearly identical, forcing the model to learn meaningful dynamics rather than relying on appearance shortcuts.

Why Does MCP Work Better at High Frame Rates? The dramatic improvement at 50 fps can be understood through supervision signal density. At high frame rates, adjacent chunks are nearly identical, making next-chunk denoising trivially solvable via appearance copying. Multi-chunk prediction breaks this shortcut: chunks 2 or 3 steps ahead exhibit substantial visual differences that can only be predicted by understanding the underlying physical dynamics, forcing the model to develop temporally-aware representations.

Table 3: Evaluation on PhyWorld benchmark for combinatorial generalization. We report FVD and Abnormal Ratio under out-of-template (OOT) and in-template (IT) settings.

	FVD (↓)		Abnormal Ratio (↓)	
Method	OOT	IT	OOT	IT
LingBot-VA	5.3	3.5	12%	3%
Next Forcing	**4.7**	**3.2**	**8%**	**2%**

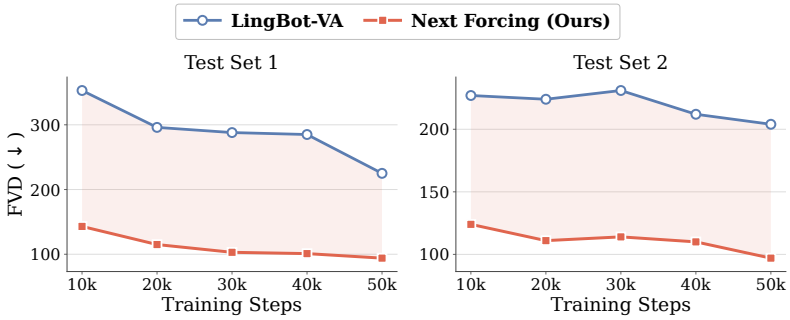


Figure 4: **FVD (↓) on general video pretraining across training steps.** Test Set 1 contains human activity videos, while Test Set 2 focuses on camera-driven scene dynamics. Next Forcing consistently achieves substantially lower FVD than LingBot-VA on both test sets throughout training.

5.2.3 PhyWorld Benchmark

To evaluate whether MCP improves temporal and physical dynamics modeling beyond robot manipulation, we additionally evaluate Next Forcing on PhyWorld [31] in a video-only setting, removing the action stream from both Next Forcing and the baseline. We report Frechet Video Distance (FVD [56], lower is better) and Abnormal Ratio (percentage of generated videos violating physical laws, lower is better).

As shown in Table 3 and Figure 3, Next Forcing improves both FVD and Abnormal Ratio over LingBot-VA, with larger gains on the out-of-template setting, suggesting that multi-chunk prediction encourages generalizable physical dynamics rather than template-specific memorization.

5.2.4 Generality on Video Pretraining

Beyond World Action Models, we further evaluate Next Forcing on action-free general video pretraining. We conduct pretraining experiments on an in-house general video dataset comprising approximately 3.5M video clips of 5–10 seconds, predominantly featuring human activities. We remove the action stream, train on 32 GPUs, and evaluate pure video generation quality using FVD [56] (lower is better). We construct two held-out test sets of 1,024 samples each: Test Set 1 contains human activity videos, while Test Set 2 focuses on camera-driven scene dynamics.

As shown in Figure 4, Next Forcing consistently achieves substantially lower FVD than LingBot-VA throughout training on both test sets. At 50k steps, Next Forcing reduces FVD by 58% on Test Set 1 (94 *vs*.225) and 52% on Test Set 2 (97 *vs*.204). Notably, Next Forcing at only 10k steps already surpasses LingBot-VA at 50k steps on both sets, demonstrating significantly faster convergence in video quality. These results confirm that multi-chunk prediction provides effective temporal supervision for general video generation, validating the generality of Next Forcing beyond the robot manipulation domain.

5.3 Ablation Studies

We conduct systematic ablation studies on both the baseline and MCP module design choices. All ablations are trained and evaluated on the RoboTwin Clean subset for 20k steps (Table 2).

Table 4: Inference acceleration. Standard inference performs separate video denoising for each chunk. MCP-accelerated inference predicts the next video chunk in parallel via the MCP module.

	12 fps		25 fps		50 fps	
Inference Mode	Clean	Random	Clean	Random	Clean	Random
Standard	94.1	93.5	92.6	91.4	91.8	90.5
MCP-accelerated (2×)	93.5	90.6	91.0	89.8	92.2	91.3

Baseline Ablation. We first ablate key design choices of the baseline LingBot-VA to understand their impact. The default baseline uses $s_{\text{main}} = 5$ with noisy history augmentation, achieving 75.6%. Removing noisy history augmentation causes a significant drop to 69.8%, as the model can shortcut the denoising task by directly copying from the clean context rather than learning meaningful dynamics. The timestep shift has a substantial effect: $s_{\text{main}} = 1$ yields 65.3%, $s_{\text{main}} = 10$ achieves the best at 78.4%, while $s_{\text{main}} = 20/25$ gives 77.6/77.2%, showing diminishing returns. Adding MCP improves performance from 75.6% to 85.8% (+10.2%).

MCP Module Ablation. From the default MCP configuration (Table 2), we ablate each design choice. Removing multi-layer fusion drops to 83.6%, confirming that fusing intermediate features enables deeper gradient propagation into the main model. Setting $s_{\text{mcp}} = 5$ (same as main) reduces to 83.2%, as the higher shift forces the MCP modules to rely more on the main model's representations and strengthens their coupling. Without weight initialization yields 83.8%. For transformer blocks, 1 block achieves 86.5% and 5 blocks gives 85.0%, suggesting that lighter MCP modules result in tighter coupling and more effective supervision. Despite the slightly higher rate with 1 block, we default to 3 blocks as it produces fewer visual artifacts in the MCP-generated chunks, which is important for the parallel chunk generation mode at inference.

5.4 Inference Acceleration

As described in Section 4.6, the MCP modules can be retained to generate two video chunks per autoregressive step, reducing the number of sequential video denoising steps. Table 4 shows a frame-rate-dependent throughput–accuracy trade-off. At 12 and 25 fps, MCP-accelerated inference reduces task success rate, with the largest drop being 2.9 percentage points on the 12 fps Random setting. At 50 fps, it matches or slightly exceeds standard inference in this evaluation. The accelerated mode is therefore more suitable at high frame rates, while standard inference can be used when strict accuracy parity is required.

6 Conclusion

We presented **Next Forcing**, a multi-chunk prediction framework for World Action Models that addresses myopic supervision in autoregressive video prediction. By improving future visual representations, Next Forcing also benefits action decoding in joint video-action architectures. By training lightweight MCP modules to predict multiple future chunks alongside the main model, Next Forcing provides dense temporal supervision that forces the model to learn long-range dynamics rather than relying on appearance shortcuts. Next Forcing establishes new state-of-the-art results on RoboTwin (94.1/93.5% on Clean/Random), achieves 2.3× faster training convergence at 50 fps, enables inference acceleration, and demonstrates over 50% FVD reduction on general video pretraining, validating its generality beyond robot-specific data and applicability to action-free video generation. The main limitation is that the MCP modules introduce extra training cost. We hope Next Forcing motivates further investigation of training objectives, beyond context construction and noise scheduling, as a key axis for improving autoregressive video generation.

Acknowledgments and Disclosure of Funding

This research is supported by the National Natural Science Foundation of China (623B2036, U25B2045, 62472184), the Fundamental Research Funds for the Central Universities, and the Innovation Project of Optics Valley Laboratory (Grant No. OVL2025YZ005).

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[2] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[3] BeingBeyond Team. Being-h0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026.

[4] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[5] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024.

[6] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

[7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

[9] Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

[10] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *International Conference on Machine Learning (ICML)*, 2024.

[11] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

[12] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[13] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[14] Kaijin Chen, Dingkang Liang, Xin Zhou, Yikang Ding, Xiaoqiang Liu, Pengfei Wan, and Xiang Bai. Out of sight but not out of mind: Hybrid memory for dynamic video world models. *arXiv preprint arXiv:2603.25716*, 2026.

[15] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[16] Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025.

[17] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[18] DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[19] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[20] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

[21] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[22] Gemini Robotics Team. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[23] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *International Conference on Machine Learning (ICML)*, 2024.

[24] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[25] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[26] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[27] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, 2025.

[28] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[29] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

[30] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

[31] Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.

[32] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

[33] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

[34] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations (ICLR)*, 2024.

[35] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, 2023.

[36] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[37] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.

[38] Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

[39] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[40] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025.

[41] Mingyu Liu, Jiuhe Shu, Hui Chen, Zeju Li, Canyu Zhao, Jiange Yang, Shenyuan Gao, Hao Chen, and Chunhua Shen. Stamo: Unsupervised learning of generalizable robot motion from compact state representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 35014–35024, 2026.

[42] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations (ICLR)*, 2025.

[43] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[44] NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[45] Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025.

[46] Physical Intelligence. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

[47] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[48] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems (RSS)*, 2025.

[49] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[50] Shuhuai Ren, Shuming Ma, Xu Sun, and Furu Wei. Next block prediction: Video generation via semi-autoregressive modeling. *arXiv preprint arXiv:2502.07737*, 2025.

[51] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. GAIA-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

[52] Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations (ICLR)*, 2024.

[53] Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025.

[54] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[55] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024.

[56] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[57] Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[58] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[59] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. TinyVLA: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024.

[60] John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. *arXiv preprint arXiv:2510.27607*, 2025.

[61] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2024.

[62] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, Lars Liden, and Jianfeng Gao. Magma: A foundation model for multimodal AI agents. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[63] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[64] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025.

[65] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22963–22974, 2025.

[66] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

[67] Hai Zhang, Siqi Liang, Li Chen, Yuxian Li, Yukuan Xu, Yichao Zhong, Fu Zhang, and Hongyang Li. Sparse video generation propels real-world beyond-the-view vision-language navigation. *arXiv preprint arXiv:2602.05827*, 2026.

[68] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[69] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

[70] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3D vision-language-action generative world model. In *International Conference on Machine Learning (ICML)*, 2024.

[71] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[72] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

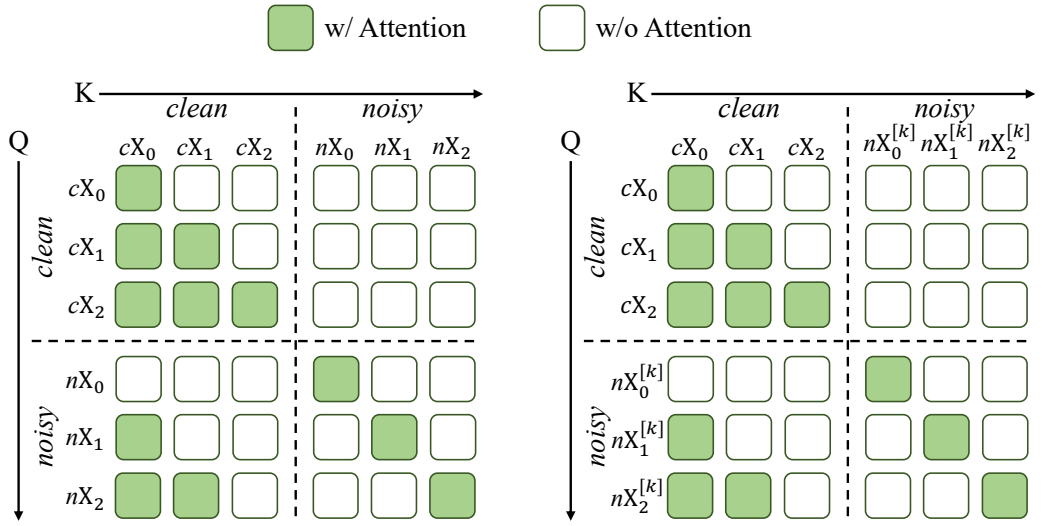
[73] Jiaming Zhou, Ke Ye, Jiayi Liu, Teli Ma, Zifan Wang, Ronghe Qiu, Kun-Yu Lin, Zhilin Zhao, and Junwei Liang. Exploring the limits of vision-language-action manipulation in cross-task generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:139899–139927, 2026.

[74] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems (RSS)*, 2025.

Supplementary Material

This appendix provides additional details that complement the main paper. Appendix A describes the attention mask design shared by the main model and MCP modules. Appendix B presents the full numerical results for the training convergence experiments. Appendix C details the timestep shift formulation used for noise scheduling.

A Attention Mask Details



(a) Attention mask for main model (b) Attention mask for MCP modules

Figure 5: **Attention mask for main model and MCP modules.** Only video tokens are shown for clarity (action tokens omitted). Under teacher forcing, the sequence consists of noisy tokens (current chunk being denoised) and clean tokens (ground-truth context). Noisy tokens attend to all causally preceding clean tokens and to noisy tokens within the same chunk; clean tokens follow a standard causal pattern; clean tokens cannot attend to noisy tokens. The MCP modules share the same attention mask structure as the main model, requiring only a single mask construction per training step.

Figure 5 illustrates the attention mask shared by both the main model and MCP modules. For clarity, only video tokens are shown (action tokens omitted). The sequence consists of two groups: noisy tokens (the current chunk being denoised) and clean tokens (ground-truth context from previous chunks). The attention rules under teacher forcing are:

- **Noisy → Clean:** Each noisy token attends to all causally preceding clean tokens, excluding the clean token at the same chunk index. This prevents information leakage from the current chunk's ground truth.
- **Noisy → Noisy:** Noisy tokens only attend to other noisy tokens within the same chunk (self-attention within the chunk being denoised).
- **Clean → Clean:** Clean tokens follow a standard causal pattern, attending to all clean tokens at the same or earlier chunk indices.
- **Clean → Noisy:** Not permitted. Clean context tokens cannot attend to noisy tokens.

A key design choice of Next Forcing is that the MCP modules reuse the same attention mask as the main model. Since both operate on sequences with the same structure (noisy target tokens + clean context tokens), the mask can be constructed once per training step and shared across the main model and all MCP depths, reducing training overhead.

B Detailed Training Convergence Results

Table 5 provides the full training convergence data at 12 fps and 50 fps, complementing the convergence curves in Figure 1. Success rates are reported every 5k training steps from 5k to 50k.

Table 5: Task success rate (%) on RoboTwin across training steps at 12 fps and 50 fps. Next Forcing consistently outperforms LingBot-VA. At 12 fps, Next Forcing leads by ∼10 points at 5k steps and maintains a 1–2 point advantage at convergence. At 50 fps, the gap is substantially larger: Next Forcing leads by 24.7/29.7 points (Clean/Random) at 5k steps, and the advantage persists through convergence.

FPS	Setting	Method	5k	10k	15k	20k	25k	30k	35k	40k	45k	50k
12	Clean	LingBot-VA	74.0	85.2	87.8	90.8	92.3	91.3	92.8	92.9	93.1	92.8
		Next Forcing	**84.9**	**90.0**	**91.5**	**92.3**	**93.3**	**94.3**	**93.1**	**93.4**	**94.2**	**94.1**
	Random	LingBot-VA	73.5	82.2	85.0	88.3	88.9	89.4	89.3	91.2	91.4	91.8
		Next Forcing	**80.6**	**85.4**	**85.8**	**90.5**	**89.8**	**91.5**	**89.6**	**91.5**	**91.6**	**93.5**
50	Clean	LingBot-VA	45.5	64.8	69.6	78.5	79.0	81.2	82.4	83.8	87.4	88.6
		Next Forcing	**70.2**	**80.5**	**85.2**	**87.4**	**87.6**	**90.0**	**90.9**	**91.5**	**91.7**	**91.8**
	Random	LingBot-VA	31.9	54.7	59.8	69.4	70.7	75.6	79.2	80.4	84.5	85.2
		Next Forcing	**61.6**	**77.6**	**80.2**	**85.0**	**85.4**	**86.8**	**89.9**	**88.4**	**89.6**	**90.5**

Several observations emerge from the detailed results. First, Next Forcing shows the largest absolute improvement in the early training stages (5k–10k steps), indicating that multi-chunk prediction provides a stronger learning signal from the start by preventing appearance shortcuts. Second, the improvement from MCP is notably frame-rate dependent. At 12 fps, adjacent chunks differ substantially in visual content, so the appearance shortcut is less severe and the baseline can still learn meaningful dynamics, resulting in a moderate improvement from MCP. At 50 fps, however, adjacent chunks are nearly identical, making it easy for the model to shortcut by simply copying the previous chunk. In this regime, the baseline struggles to learn beyond local copying, while MCP forces the model to capture long-range dynamics, leading to dramatically larger gains.

C Timestep Shift Formulation

Here we provide the detailed formulation of the timestep shift mechanism used in both the main model and MCP modules.

Shifted Timestep Schedule. We first construct a schedule of $T = 1000$ uniformly spaced base values $\{\sigma_i\}_{i=0}^{T-1}$ over $[\sigma_{\min}, \sigma_{\max}] = [0, 1]$. These are then transformed by the shift parameter s [20]:

$$\tilde{\sigma}_i = \frac{s \cdot \sigma_i}{1 + (s-1) \cdot \sigma_i}. \tag{14}$$

This transformation is monotonic and maps $[0, 1] \to [0, 1]$, but redistributes the noise levels across the interval: a larger s pushes $\tilde{\sigma}$ toward higher values, concentrating training on noisier regimes. For the main model we use $s_{\text{main}} = 5$; for the MCP modules we use $s_{\text{mcp}} = 10$.

Timestep Sampling. During training, a timestep index is uniformly sampled: $\text{id} \sim \text{Uniform}\{0, 1, \ldots, T-1\}$, and the corresponding shifted noise level $\tilde{\sigma}_{\text{id}}$ is looked up from the pre-computed schedule. The noisy training sample is then constructed as:

$$\mathbf{x}_{\tilde{\sigma}} = (1 - \tilde{\sigma}_{\text{id}})\, \mathbf{x}_0 + \tilde{\sigma}_{\text{id}}\, \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \tag{15}$$

Since the shift is already encoded in the schedule, uniform sampling over the shifted schedule effectively produces a non-uniform distribution over noise levels, biased toward higher noise for larger shift s.

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: The abstract and introduction clearly state three main contributions: (1) the Next Forcing framework for overcoming myopic supervision (Section 4), (2) faster training convergence and new state-of-the-art results on RoboTwin and PhyWorld (Section 5), and (3) $2\times$ inference acceleration (Section 4.6). All claims are supported by experimental results.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We discuss limitations in the conclusion section. The MCP modules add approximately 15% additional parameters during training compared to the baseline, introducing moderate training overhead.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: This paper is primarily empirical and does not include theoretical results or proofs.

Guidelines:

- The answer [N/A] 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 provide detailed implementation and training details in Section 5.

Guidelines:

- The answer [N/A] 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: [No]

   Justification: We plan to release the code upon acceptance. The evaluation benchmarks RoboTwin and PhyWorld are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and evaluation details are provided in Section 5, including data splits (2,500 Clean + 25,000 Random demonstrations), hyperparameters (s_{main}, s_{mcp}, loss weights, chunk size), and evaluation metrics (task success rate, FVD, Abnormal Ratio).

   Guidelines:

   - The answer [N/A] 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: We follow the standard evaluation protocol of the RoboTwin benchmark, which reports average success rates over all 50 tasks. Error bars are not reported as each task is evaluated with a fixed number of trials following the benchmark convention, consistent with prior work.

   Guidelines:

   - The answer [N/A] 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: Training is conducted on 64 NVIDIA H20 GPUs. The MCP modules add approximately 15% parameter and compute overhead during training compared to the baseline.

   Guidelines:

   - The answer [N/A] 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: This research conforms with the NeurIPS Code of Ethics. The work focuses on improving training objectives for video world models in robot manipulation, using publicly available simulation benchmarks.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: This work focuses on improving training objectives for autoregressive video generation. It is foundational research not tied to specific applications, and we do not foresee direct societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Our method is a training objective for robot manipulation and does not pose significant risks for misuse.

Guidelines:

- The answer [N/A] 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: We properly cite the RoboTwin benchmark, PhyWorld benchmark, LingBot-VA framework, and the Wan2.2 backbone model. All used assets are publicly available research artifacts.

Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: This paper does not release new datasets or pre-trained models. We plan to release the code upon acceptance.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

Justification: This paper does not involve research with human subjects.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the core method in this research.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.