# Action-Controlled Scale-Wise Flow Matching for Embodied World Models

Sen Wang, Sanping Zhou, *Member, IEEE,* Huaiyi Dong, Kun Xia, *Member, IEEE,* Gang Hua, *Fellow, IEEE,* and Le Wang, *Senior Member, IEEE*

***Abstract*—Action-conditioned world models are useful for embodied agents only when their predicted futures remain controllable by actions and stable under long-horizon rollout. We present SAMPO++, an embodied world model that treats action-conditioned prediction as a scale-decoupled controlled dynamical system rather than as conventional action-conditioned video generation. SAMPO++ couples temporal autoregression with scale-wise flow matching in a continuous latent pyramid. A multi-scale temporal planner summarizes the latent history with scale-matched recurrent states, allowing dynamics at different resolutions to be conditioned by appropriate temporal contexts. An Action-Controlled Velocity Field (ACVF) separates action-free passive drift from action-induced residual dynamics, making the action an explicit bounded control input to the flow velocity instead of a passive conditioning token. To improve cross-scale consistency and closed-loop stability, SAMPO++ further uses Pyramid-Consistent RoPE (PC-RoPE) and rollout-aware training, aligning scale-wise coordinates and exposing the model to its own autoregressive prediction distribution during training. Beyond standard perceptual metrics such as FVD, PSNR, SSIM, and LPIPS, we evaluate world-model-native properties including action alignment, counterfactual accuracy, no-op residual, and rollout drift. We further study SAMPO++ as a learned simulator for visual planning and policy-level evaluation, where policies or action chunks can be rolled out and compared before execution. Experiments across action-conditioned robotic manipulation, visual planning, and model-based reinforcement learning, together with action-free driving video prediction, show that SAMPO++ improves visual prediction quality while providing stronger action alignment, counterfactual accuracy, no-op residual suppression, and long-horizon rollout consistency over strong discrete and continuous baselines. Videos and additional details are available at project page.**

***Index Terms*—Action-Conditioned World Modeling, Learned Controllable Simulation, Continuous Representation Learning, Flow Matching, Embodied AI.**

Manuscript received xxx; revised xxx; accepted xxx. Date of publication xxx; date of current version xxx. This work was supported in part by the National Key Research and Development Project under Grant 2024YFB4708100, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM504, National Natural Science Foundation of China under Grants 62572384, U24A20325, 62125305, 62503384, and Key Research and Development Plan of Shaanxi Province under Grant 2024PT-ZCK-80. (*Corresponding author: Sanping Zhou.*)

Sen Wang, Sanping Zhou, Huaiyi Dong and Le Wang are with the State Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: sen.wang@stu.xjtu.edu.cn; spzhou@mail.xjtu.edu.cn; Donghuaiyi@stu.xjtu.edu.cn; lewang@mail.xjtu.edu.cn;). Kun Xia is also with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: kunxia@mail.xjtu.edu.cn).

Gang Hua is with Amazon Alexa AI, Bellevue, WA 98004 USA (e-mail: ganghua@gmail.com).

## I. Introduction

WORLD models enable embodied agents to predict the future consequences of their actions before acting in the environment [1]–[3]. In robotics, visual planning, autonomous driving, and model-based reinforcement learning, such predictive models are expected to serve not only as high-fidelity video generators but also as controllable transition models that remain reliable over long horizons [4]–[8]. This distinction is central: a plausible future video is not necessarily a useful world-model rollout. For embodied decision making, the predicted future must change consistently with the executed action, preserve relevant objects and contacts, and avoid compounding drift when predictions are fed back autoregressively.

A reliable action-conditioned world model can also act as a learned simulator. Given a policy or a set of candidate action sequences, the model can roll out possible futures, estimate task progress, and rank alternatives before physical execution. This is particularly valuable when real-world trials are expensive, unsafe, or slow. However, this simulator role raises a stronger requirement than open-loop video prediction: the rollout must remain action-sensitive, discriminative under action replacement, and stable under feedback. A model that produces visually plausible frames but ignores the action may score well under perceptual metrics while being unusable for policy evaluation or action-sequence selection.

Existing visual world models have made substantial progress, yet their design often inherits assumptions from video generation rather than controlled dynamics. Discrete autoregressive models tokenize visual observations with VQ-VAE or VQGAN codebooks [9], [10] and exploit Transformer-based causal sequence modeling [11]–[13]. These models provide a natural autoregressive interface, but quantization can limit fine-grained fidelity and smooth physical evolution [14], [15]. Continuous diffusion and flow-based models avoid codebook artifacts and have achieved strong perceptual synthesis quality [16]–[21]. However, high visual fidelity alone does not guarantee action controllability: actions are often injected through concatenation, cross-attention, or global modulation, where they can be treated as weak conditions rather than as control variables governing the transition dynamics [22], [23].

We argue that the main obstacle is not simply the choice between discrete and continuous representations. The deeper issue is that action-conditioned world models are frequently optimized as conditional video generators rather than as con-

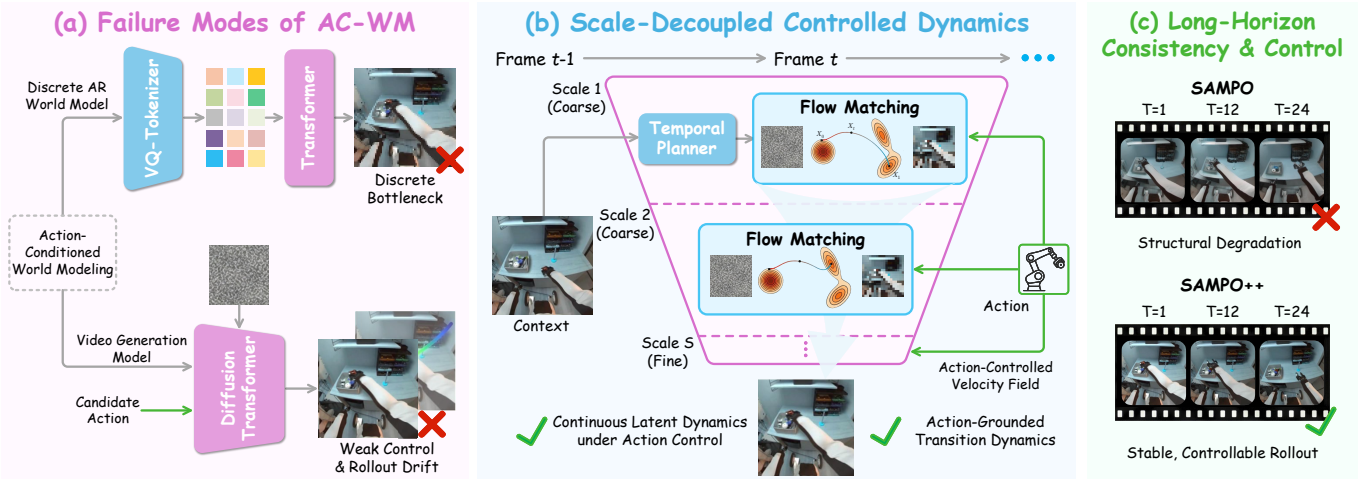


Fig. 1: **From action-conditioned video generation to controlled world modeling.** **(a)** Standard action-conditioned video models may fail as world models when actions are treated as passive conditions, multi-scale dynamics are collapsed into a single recurrent state, or teacher-forced training drifts under autoregressive rollout. **(b)** SAMPO++ addresses these failure modes with an Action-Controlled Velocity Field (ACVF), a multi-scale temporal planner, Pyramid-Consistent RoPE (PC-RoPE), and rollout-aware training. **(c)** We evaluate the resulting model not only by perceptual fidelity, but also by action alignment, counterfactual accuracy, no-op residual, rollout drift, and policy-level evaluation inside the learned simulator.

trolled dynamical systems. This leads to three failure modes. First, *passive action conditioning* can allow the model to explain future frames from visual inertia and dataset priors while ignoring the action. Second, *single-scale temporal recurrence* asks one recurrent state to summarize global object displacement, local contact dynamics, and high-frequency appearance changes, although these phenomena live at different spatial scales. Third, *teacher-forced training* exposes the model mainly to clean histories, whereas inference requires closed-loop rollout on its own imperfect predictions, producing distribution shift and long-horizon drift.

To address these limitations, we propose SAMPO++, an action-controlled scale-wise flow model for embodied world modeling. SAMPO++ combines temporal autoregression with scale-wise flow matching in a continuous latent pyramid. The temporal component provides an autoregressive recurrent interface, while the scale-wise flow component generates continuous latent states without discrete codebook restriction. Unlike standard hybrid AR-flow generators, SAMPO++ is organized around a controlled-dynamics factorization: temporal history is summarized at multiple scales, and the current action controls the generative velocity field through an explicit residual transition.

Concretely, SAMPO++ introduces three design choices. First, a *multi-scale temporal planner* reasons over the latent pyramid and emits one scale-matched recurrent state per resolution. This prevents the planner from relying only on the finest scale and gives each flow stage a temporal context appropriate to its spatial resolution. Second, an *Action-Controlled Velocity Field* decomposes the flow velocity into action-free passive drift and action-induced residual dynamics. The residual branch is bounded, spatially masked, zero-initialized, and trained with no-op and counterfactual objectives, so that the action acts as an explicit control input rather than as a decorative condition. Third, *Pyramid-Consistent RoPE* places tokens from different scales into a shared resolution-normalized coordinate system with scale-specific frequency bands, while rollout-aware training reduces the mismatch between teacher-forced learning and closed-loop autoregressive inference.

These design choices make SAMPO++ suitable not only for visual forecasting but also for policy-facing uses of world models. During planning or policy evaluation, candidate action sequences can be rolled out in the learned simulator and scored by task progress, predicted state quality, or downstream rewards. In this setting, action alignment and counterfactual accuracy are not auxiliary diagnostics; they are prerequisites for trusting the simulator's ranking. No-op residual and rollout drift further measure whether policy rollouts remain meaningful as the model repeatedly conditions on its own predictions.

We evaluate SAMPO++ across robotic manipulation datasets such as BAIR [24] and RoboNet [25], action-free driving video prediction on Cityscapes [26] and OpenDV-YouTube [27], visual planning on the VP$^2$ benchmark [28], and model-based reinforcement learning on Meta-World [29]. In addition to standard perceptual metrics, we introduce world-model-native diagnostics that directly measure action alignment, counterfactual accuracy, no-op residual, and long-horizon rollout drift. We further evaluate whether the learned rollouts can serve as a learned simulator for preliminary policy screening and action-sequence comparison. These metrics are essential because a visually plausible rollout can still be useless for planning if it is insensitive to action or unstable under feedback.

In summary, our contributions are as follows:

- We reframe action-conditioned world modeling as a scale-decoupled action-controlled transition problem with explicit closed-loop stability requirements, and instantiate this view in SAMPO++, a continuous-latent framework

that couples temporal autoregression with scale-wise flow matching.
- We introduce a multi-scale temporal planner whose recurrent state spans the latent pyramid and provides scale-matched temporal contexts for coarse-to-fine flow generation.
- We propose the Action-Controlled Velocity Field, which separates passive drift from action-induced residual dynamics and trains the residual control branch with no-op and counterfactual objectives.
- We develop Pyramid-Consistent RoPE and rollout-aware training for cross-scale coordinate consistency and long-horizon stability, and evaluate the model with both perceptual metrics and world-model-native diagnostics across robotics, action-free driving prediction, planning, learned-simulator policy evaluation, and MBRL.

This paper substantially extends our preliminary conference version [8]. The conference model performs discrete next-scale prediction with motion prompts. In contrast, SAMPO++ moves the pipeline to continuous latent flow matching and recasts the problem as controlled dynamics rather than discrete token generation. It further replaces the single-state temporal planner with a multi-scale planner, replaces affine action conditioning with ACVF and its no-op/counterfactual objectives, replaces scale-aware positional coding with PC-RoPE, and addresses long-horizon drift with rollout-aware training instead of relying on inference-time optical-flow rectification. We also add world-model-native metrics, policy-evaluation diagnostics for learned simulation, expanded long-horizon experiments, revised ablations, and a public implementation with configuration flags that preserve the conference model as a reproducible baseline.

The remainder of this paper is organized as follows. Sec. II reviews related work on autoregressive visual generation, continuous flow-based modeling, and action-conditioned world models. Sec. III presents the SAMPO++ framework. Sec. IV reports experimental results and ablation studies. Sec. V concludes the paper.

## II. Related Work

### A. Autoregressive and Scale-wise Visual Generation

Autoregressive visual models compress images or videos into latent tokens and predict them with causal sequence models [9]–[12], [15]. Their causal factorization provides a natural interface for long-horizon rollout, but conventional raster-scan orderings are inefficient for images and ignore two-dimensional spatial structure. Visual Autoregressive modeling (VAR) [30] and our conference framework SAMPO [8] address this by predicting visual representations scale by scale, generating coarse structures before finer details. This next-scale formulation shortens the effective sequence and better matches the hierarchical organization of visual signals. A limitation of many scale-wise autoregressive systems is their reliance on discrete codebooks. Quantization can introduce reconstruction artifacts and make smooth physical dynamics harder to represent. Recent work has therefore explored continuous autoregressive generation, including MAR [31] and FlowAR [32], which replace discrete token prediction with continuous diffusion- or flow-style generation. We build on this progress and do not claim novelty in simply removing quantization. For SAMPO++, the continuous scale-wise backbone is a means to a different end: it enables actions to control a smooth velocity field and allows temporal recurrence to operate on continuous multi-scale states. Our contribution is the controlled-dynamics design built on top of scale-wise continuous generation, rather than the continuous representation alone.

### B. Flow Matching and Continuous Latent Dynamics

Diffusion models and latent diffusion models learn continuous generative processes by reversing a noising trajectory and have achieved strong perceptual synthesis quality [16], [17], [21], [33], [34]. Flow Matching (FM) and Rectified Flow provide an alternative continuous formulation by learning a velocity field that transports samples from a simple source distribution to the data distribution [18], [19], [35]. These objectives often support efficient deterministic sampling and an ODE-based view of generation. Scalable interpolant and flow-based Transformer variants further show that continuous flow objectives can be combined with modern high-capacity architectures [36], [37]. Recent hybrid models combine autoregressive structure with continuous generation. Transfusion [38] and Show-o [39] study mixed discrete-continuous multimodal modeling, while MAR [31] and FlowAR [32] use diffusion or flow heads to generate continuous visual tokens under autoregressive conditioning. These methods establish that continuous autoregression can be an effective image-generation paradigm. The focus of SAMPO++ is orthogonal to image fidelity alone: embodied world modeling requires a controllable transition operator, action-sensitive dynamics, and stable closed-loop rollout. We therefore use flow matching not merely as a better decoder, but as a vehicle for an action-controlled velocity field whose residual component is explicitly tied to the executed action.

### C. Action-Conditioned World Models for Policy Evaluation

World models learn predictive simulators that support planning and policy learning [1], [2]. Recurrent state-space models such as Dreamer and its successors achieve high sample efficiency in reinforcement learning by learning compact latent dynamics [4], [5]. These methods show that learned dynamics can improve decision making, but compact latent reconstructions can be limiting for fine-grained manipulation and visually grounded planning. More recent generative world models use diffusion or Transformer backbones to synthesize higher-fidelity futures. Genie and Genie-2 learn interactive controllable environments from video [6], [40], while GameNGen demonstrates that a diffusion model can operate as a real-time game simulator [41]. These systems highlight the promise of generative world models as interactive simulators. A learned simulator becomes useful for policy evaluation only if its rollouts are sensitive to the policy's actions and stable under repeated feedback. In policy screening, action-sequence ranking, and model-based planning, the model must

distinguish whether different candidate actions lead to different task outcomes. A visually realistic rollout that ignores the action can mis-rank policies, while a model that drifts under autoregressive feedback can overestimate long-horizon success. This motivates evaluating world models not only by perceptual fidelity but also by action alignment, counterfactual consistency, no-op stability, and rollout drift. SAMPO++ is designed for this use case: candidate policies or action chunks can be rolled out in the learned controllable simulator, and their predicted futures can be compared before execution. For embodied control, however, visual realism is insufficient. Many action-conditioned video models inject actions through concatenation, cross-attention, or global conditioning [22], [23]. Such mechanisms can improve conditional generation but do not by themselves ensure that the predicted transition is causally aligned with the action, especially under long rollout. In parallel, feature-space predictive approaches such as JEPA reduce visual stochasticity by predicting abstract representations [42], [43], whereas full generative world models synthesize observable futures that are easier to inspect and use for visual planning. SAMPO++ follows the generative direction but places action controllability at the level of the transition operator: the action modulates the flow velocity through a bounded residual control branch on top of passive drift. Long-horizon stability is another key challenge. Teacher-forced training evaluates transitions from clean histories, whereas deployment repeatedly conditions on generated states. This mismatch can cause object disappearance, action drift, and accumulated visual artifacts. Existing work often addresses temporal consistency through stronger architectures, positional encodings, or inference-time guidance. SAMPO++ instead treats closed-loop stability as part of the training and representation design: PC-RoPE aligns scale-wise coordinates in a shared physical frame, while rollout-aware training exposes the model to its own predicted histories. Together with ACVF and world-model-native evaluation, this yields a framework aimed at action-controllable and long-horizon-stable embodied world modeling rather than only photorealistic video prediction.

## III. Methodology

### A. Problem Statement

We study action-conditioned world modeling for embodied agents [1], [2], [5], [44]. Let $\boldsymbol{o}_t$ denote the RGB observation at video time $t$, and let $\boldsymbol{a}_t$ denote the action executed between time $t$ and time $t+1$. We use a next-state convention throughout the paper: $\boldsymbol{a}_t$ is the control input that induces the transition from $\boldsymbol{o}_t$ to $\boldsymbol{o}_{t+1}$. This convention matches policy-facing use cases: a policy proposes an action, the world model predicts the next state, and the policy can then be queried again on the predicted observation.

SAMPO++ operates in a continuous pyramidal latent space rather than directly in pixels [17], [31], [45]. Each observation is encoded as

$$\boldsymbol{z}_t = \mathcal{E}(\boldsymbol{o}_t) = \{\boldsymbol{z}_t^s\}_{s=1}^{S}, \quad \boldsymbol{z}_t^s \in \mathbb{R}^{H_s \times W_s \times C_s}, \tag{1}$$

where $\mathcal{E}$ is the visual tokenizer, $S$ is the number of spatial scales, $s=1$ denotes the coarsest scale, $s=S$ denotes the finest scale, and $(H_s, W_s, C_s)$ are the spatial resolution and channel dimension of scale $s$. When visual rollouts are required, a decoder $\mathcal{D}$ maps the generated latent pyramid back to the observation space. Unless otherwise stated, the tokenizer is frozen during world-model training, which fixes the latent state space and keeps the comparison with the conference version controlled.

The transition model estimates the next latent pyramid under the current action and the historical context:

$$p_\theta(\boldsymbol{z}_{t+1}^{1:S} \mid \boldsymbol{z}_{\le t}^{1:S}, \boldsymbol{a}_{\le t}). \tag{2}$$

Equation (2) is the basic object used for open-loop future prediction, visual planning, and closed-loop policy rollout [2], [24], [44], [46]. When rewards or task-success labels are available, they are predicted by a separate reward or success head introduced in Sec. III-F; this head is used only for policy scoring and is not part of the visual prediction objective.

This formulation exposes three requirements that are not captured by perceptual video metrics alone [28], [47], [48]. First, the model must be action-sensitive: different actions from the same state should lead to distinguishable futures. Second, the model must be scale-aware: global displacement, contact, articulation, and appearance residuals occur at different spatial frequencies. Third, the model must be stable under closed-loop rollout: at inference time, it consumes its own predictions rather than ground-truth histories. SAMPO++ addresses these requirements with a multi-scale temporal planner, a scale-wise conditional flow renderer, an Action-Controlled Velocity Field (ACVF), Pyramid-Consistent RoPE (PC-RoPE), and rollout-aware training. Fig. 2 gives the overall architecture.

### B. Continuous Pyramidal Latent Representation

SAMPO++ represents each frame as a hierarchy of continuous latent grids, following the scale-wise generative structure of recent next-scale models [8], [30] while adapting it to action-conditioned temporal dynamics. Coarse scales encode global layout and large object displacement, whereas finer scales preserve contact regions, local deformation, and appearance details. We do not present the continuous latent space itself as the sole novelty of SAMPO++; rather, it provides the differentiable state space on which controlled flow dynamics can be defined.

In practice, the visual tokenizer first maps an observation to a base latent feature map $\boldsymbol{f}_t^0$. A pyramidal downsampling operator constructs the scale hierarchy:

$$\boldsymbol{z}_t^S = \boldsymbol{f}_t^0, \quad \boldsymbol{z}_t^{s-1} = P_\downarrow(\boldsymbol{z}_t^s), \quad s = S, \dots, 2, \tag{3}$$

where $P_\downarrow$ compresses spatial resolution while preserving the continuous channel representation. The inverse resizing operator $P_\uparrow$ is used only when a coarser latent is spatially aligned as a condition for a finer-scale renderer.

Within a frame, the next latent pyramid is generated from coarse to fine. We denote by

$$\boldsymbol{z}_{t+1}^{<s} \triangleq \{\boldsymbol{z}_{t+1}^k\}_{k=1}^{s-1} \tag{4}$$

the set of all coarser scales of the next frame that have already been generated. The temporal planner produces one

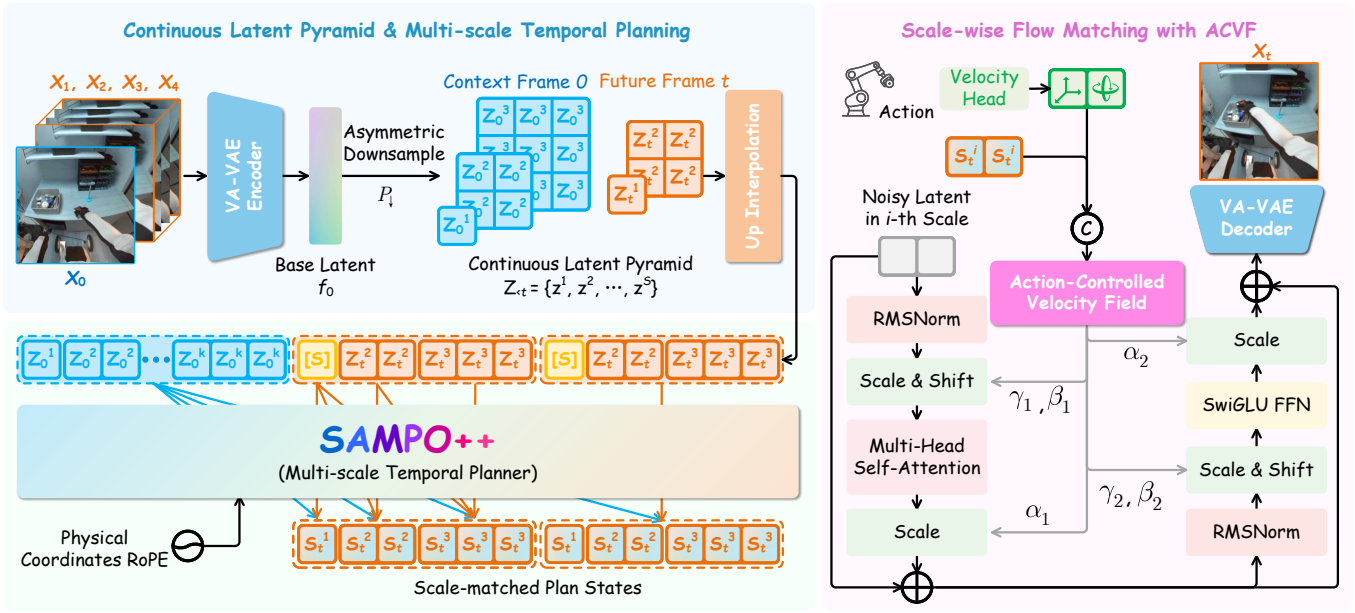


Fig. 2: **Overview of SAMPO++.** SAMPO++ encodes context observations into continuous latent pyramids, summarizes the history with a multi-scale temporal planner, and generates the next frame from coarse to fine using action-controlled scale-wise flow matching.

scale-matched recurrent state for each scale, and the renderer factorizes the next-frame distribution as

$$\begin{aligned} \{\boldsymbol{h}_{t+1}^{s}\}_{s=1}^{S} &= \Phi_{\theta}^{\text{time}}(\boldsymbol{z}_{\le t}^{1:S}, \boldsymbol{a}_{<t}), \\ p_{\theta}(\boldsymbol{z}_{t+1}^{1:S} \mid \boldsymbol{z}_{\le t}^{1:S}, \boldsymbol{a}_{\le t}) &= \prod_{s=1}^{S} p_{\theta}(\boldsymbol{z}_{t+1}^{s} \mid \boldsymbol{z}_{t+1}^{<s}, \boldsymbol{h}_{t+1}^{s}, \boldsymbol{a}_{t}), \end{aligned} \tag{5}$$

where $\Phi_{\theta}^{\text{time}}$ is the temporal planner and $\boldsymbol{h}_{t+1}^{s}$ is the recurrent state used by scale $s$ of the next transition. The planner summarizes the observation history and past actions only; the current action $\boldsymbol{a}_t$ is injected by the renderer through ACVF. This separation avoids routing the current control command through a passive history pathway and makes the role of the current action explicit in the velocity field. During training, the coarser-scale condition $\boldsymbol{z}_{t+1}^{<s}$ can be taken from the encoded ground truth or from scheduled generated latents. During inference, the scales are always generated recursively from $s = 1$ to $S$.

### C. Action-Controlled Scale-Wise Flow Matching

SAMPO++ decomposes action-conditioned world modeling into two coupled components. The inter-frame temporal planner summarizes the multi-scale history into compact scale-matched states. The intra-frame scale-wise flow renderer transforms Gaussian noise into each latent scale under the corresponding temporal state and current action. This section introduces the planner and renderer before describing the action-control parameterization used inside the velocity network.

**Multi-scale temporal planner.** The planner receives the latent history $\boldsymbol{z}_{\le t}^{1:S}$ and the past action history $\boldsymbol{a}_{<t}$. Each latent map is projected to a shared hidden dimension and augmented with temporal, spatial, and scale information. A causal Transformer processes the resulting sequence [11], and a scale-specific readout extracts a compact recurrent state:

$$\boldsymbol{h}_{t+1}^{s} = \text{Readout}_{s}\left(\Phi_{\theta}^{\text{time}}(\text{Embed}(\boldsymbol{z}_{\le t}^{1:S}, \boldsymbol{a}_{<t}))\right), \quad \boldsymbol{h}_{t+1}^{s} \in \mathbb{R}^{D}, \tag{6}$$

where $D$ is the planner hidden dimension, $\text{Embed}(\cdot)$ denotes the input embedding function, and $\text{Readout}_s$ is the scale-specific readout for scale $s$. The state $\boldsymbol{h}_{t+1}^{s}$ is intentionally compact: it is a scale-matched recurrent summary rather than a dense spatial planning map. This design keeps the temporal planner efficient while avoiding a single global state for all spatial frequencies; the subsequent flow renderer remains responsible for spatially detailed generation.

To ensure that the planner state carries predictive information before flow-based refinement, we use a lightweight scale-specific prediction head. Because $\boldsymbol{h}_{t+1}^{s}$ is a compact state rather than a dense map, the head predicts a pooled latent target instead of the full spatial grid. Let $\text{Pool}_s(\cdot)$ denote spatial pooling followed by a projection to the planner dimension, and define

$$\bar{\boldsymbol{z}}_{t+1}^{s} = \text{Pool}_{s}(\boldsymbol{z}_{t+1}^{s}). \tag{7}$$

The auxiliary planning objective is

$$\begin{aligned} \tilde{\boldsymbol{z}}_{t+1}^{s} &= g_{s}(\boldsymbol{h}_{t+1}^{s}), \\ \mathcal{L}_{\text{plan}} &= \sum_{s=1}^{S} \alpha_{s} \left\| \tilde{\boldsymbol{z}}_{t+1}^{s} - \text{sg}(\bar{\boldsymbol{z}}_{t+1}^{s}) \right\|_{2}^{2}, \end{aligned} \tag{8}$$

where $\tilde{\boldsymbol{z}}_{t+1}^{s} \in \mathbb{R}^{D}$ is the predicted pooled latent state, $\alpha_s$ is the loss weight for scale $s$, and $\text{sg}(\cdot)$ denotes stop-gradient. The auxiliary loss in Eq. (8) does not ask the planner to synthesize spatial details; it only encourages the compact state to encode next-step temporal information before flow-based refinement.

**Scale-wise conditional flow renderer.** For each scale $s$, the renderer samples $\boldsymbol{z}_{t+1}^{s}$ from Gaussian noise conditioned on $\boldsymbol{z}_{t+1}^{<s}$, $\boldsymbol{h}_{t+1}^{s}$, and $\boldsymbol{a}_t$. Let $\tau \in [0,1]$ denote flow time, which is distinct from video time $t$. We define the noise endpoint $\boldsymbol{x}_0^s$, the data endpoint $\boldsymbol{x}_1^s$, and the linear interpolation state as

$$\boldsymbol{x}_0^s \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \boldsymbol{x}_1^s = \boldsymbol{z}_{t+1}^s, \quad \boldsymbol{x}_\tau^s = (1-\tau)\boldsymbol{x}_0^s + \tau \boldsymbol{x}_1^s. \tag{9}$$

The corresponding optimal-transport target velocity is

$$\boldsymbol{v}^{\star,s} = \boldsymbol{x}_1^s - \boldsymbol{x}_0^s. \tag{10}$$

The neural velocity field at scale $s$ is written as the probability-flow ODE

$$d\boldsymbol{x}_\tau^s = v_\theta^s(\boldsymbol{x}_\tau^s, \tau; \boldsymbol{z}_{t+1}^{<s}, \boldsymbol{h}_{t+1}^s, \boldsymbol{a}_t) d\tau. \tag{11}$$

The renderer is trained by the conditional flow-matching regression objective [18], [19], [45], [49]

$$\mathcal{L}_{\text{Flow}} = \mathbb{E}_{t,s,\tau,\boldsymbol{x}_0,\boldsymbol{x}_1}\left[ w_s \left\| v_\theta^s(\boldsymbol{x}_\tau^s, \tau; \boldsymbol{z}_{t+1}^{<s}, \boldsymbol{h}_{t+1}^s, \boldsymbol{a}_t) - \boldsymbol{v}^{\star,s} \right\|_2^2 \right], \tag{12}$$

where $w_s$ balances the contribution of different scales. The conditioning variables are written explicitly to avoid ambiguity between video time $t$, flow time $\tau$ and scale index $s$.

### D. Action-Controlled Velocity Field

Existing action-conditioned video models often inject the action as a concatenated token, a cross-attention condition, or an affine modulation [6], [7], [22], [23], [50]. These designs are straightforward, but they do not explicitly separate action-independent drift from action-induced change. In long rollouts, passive conditioning can be diluted by visual context or can globally perturb features unrelated to the executed action. SAMPO++ instead parameterizes the velocity network with an explicit passive branch and an action-residual branch, which we call the Action-Controlled Velocity Field (ACVF).

We use the centered action $\bar{\boldsymbol{a}}_t = \boldsymbol{a}_t - \boldsymbol{a}_\varnothing$, where $\boldsymbol{a}_\varnothing$ is the dataset-specific no-op command in the normalized action space. For velocity or delta-pose control, $\boldsymbol{a}_\varnothing$ is typically the zero command. For datasets without an explicit no-op, we estimate $\boldsymbol{a}_\varnothing$ from stationary segments and otherwise disable no-op-specific diagnostics.

For a hidden feature $\boldsymbol{u}$ at scale $s$ and transition $t \to t+1$, ACVF decomposes the predicted velocity into a passive branch and an action-control residual:

$$\boldsymbol{v}_\theta^s = \underbrace{\boldsymbol{v}_{\text{drift}}^s(\boldsymbol{u}; \boldsymbol{c}_{t+1,s})}_{\text{passive branch}} + \underbrace{\eta_s \boldsymbol{m}_{t+1,s} \odot (\boldsymbol{g}_{t+1,s} \odot \boldsymbol{\Phi}_s(\boldsymbol{u}, \bar{\boldsymbol{a}}_t))}_{\text{action-control residual}}, \tag{13}$$

where $\boldsymbol{c}_{t+1,s}$ denotes action-free structural conditions, including $\boldsymbol{z}_{t+1}^{<s}$, $\boldsymbol{h}_{t+1}^s$, and flow-time embeddings. The scalar $\eta_s$ is a learnable residual gate initialized at zero, $\boldsymbol{m}_{t+1,s}$ is a spatial action-influence mask, and $\boldsymbol{g}_{t+1,s}$ is a bounded channel gain. The passive branch can model action-independent dynamics, such as inertia or background motion; the residual branch captures the part of the velocity that depends on the current control input.

The control residual is computed from the hidden feature and the centered action:

$$\boldsymbol{\Phi}_s(\boldsymbol{u}, \bar{\boldsymbol{a}}_t) = \boldsymbol{W}_o\left(\text{RMSNorm}(\boldsymbol{u}) \odot \boldsymbol{B}_s \psi(\bar{\boldsymbol{a}}_t)\right), \tag{14}$$

where $\psi(\cdot)$ is an action encoder satisfying $\psi(\mathbf{0}) = \mathbf{0}$, $\boldsymbol{B}_s$ is a small-random-initialized scale-specific action basis, and $\boldsymbol{W}_o$ is a bias-free output projection. The channel gain and spatial mask are

$$\begin{aligned} \boldsymbol{g}_{t+1,s} &= \gamma_{\max} \tanh(\boldsymbol{W}_g \boldsymbol{h}_{t+1}^s), \\ \boldsymbol{m}_{t+1,s} &= \sigma\left(\text{MLP}([\text{RMSNorm}(\boldsymbol{u}), \boldsymbol{h}_{t+1}^s])\right), \end{aligned} \tag{15}$$

where $\gamma_{\max}$ controls the maximum modulation amplitude, $\boldsymbol{W}_g$ is a learned projection, and $\sigma(\cdot)$ is the sigmoid function. The same passive/action split is applied to the final velocity prediction head, so that $\boldsymbol{v}_{\text{drift}}$ and $\Delta \boldsymbol{v}_{\text{ctrl}}$ are explicitly available for no-op residual, counterfactual accuracy, and per-scale action-energy diagnostics.

This construction differs from affine conditioning in three concrete ways. First, the action contributes an additive residual on top of an action-free branch rather than globally shifting all features. Second, the residual is spatially localized and scale-specific. Third, when $\bar{\boldsymbol{a}}_t = \mathbf{0}$, the control residual is structurally zero, while the passive branch may still predict action-independent scene dynamics. This does not imply a static transition under the no-op command; only the action-induced residual is constrained to vanish.

**Counterfactual action objective.** To make the action pathway predictive rather than merely correlational [28], [51], we add a counterfactual action-discrimination objective. For each factual action $\boldsymbol{a}_t$, we sample a negative action $\tilde{\boldsymbol{a}}_t$ from temporally mismatched or in-batch actions. To avoid false negatives, we keep the pair only when the action distance exceeds $\delta_a$ and the observed pooled latent displacement exceeds $\delta_z$. The pooled latent $\bar{\boldsymbol{z}}_t^s$ is defined by the same pooling operator used in Eq. (7). This validity mask is

$$\mathbf{1}_{t,s}^{\text{cf}} = \mathbf{1}[\|\boldsymbol{a}_t - \tilde{\boldsymbol{a}}_t\|_2 > \delta_a]\, \mathbf{1}\left[\|\bar{\boldsymbol{z}}_{t+1}^s - \bar{\boldsymbol{z}}_t^s\|_2 > \delta_z\right]. \tag{16}$$

This mask avoids penalizing visually indistinguishable action pairs or transitions with negligible observed motion. The velocity induced by the factual action should match the observed transition better than the velocity induced by a mismatched action:

$$\mathcal{L}_{\text{cf}} = \mathbb{E}\left[ \mathbf{1}_{t,s}^{\text{cf}} \left[ \rho + \|v_\theta^s(\cdot, \boldsymbol{a}_t) - \boldsymbol{v}^{\star,s}\|_2^2 - \|v_\theta^s(\cdot, \tilde{\boldsymbol{a}}_t) - \boldsymbol{v}^{\star,s}\|_2^2 \right]_+ \right], \tag{17}$$

where $\rho$ is a margin, and $[\cdot]_+$ denotes the hinge function. The dot notation in $v_\theta^s(\cdot, \boldsymbol{a})$ means that the noisy state, flow time, coarser-scale context, and planner state are kept fixed while only the action input is replaced. In practice, the counterfactual objective is applied mainly to coarse and middle scales, where action-induced displacement and contact dynamics are more reliable than high-frequency appearance residuals. We later report the per-scale action energy $E_s(\boldsymbol{a}) = \|v_\theta^s(\cdot, \boldsymbol{a}) - v_\theta^s(\cdot, \boldsymbol{a}_\varnothing)\|_2$ as a diagnostic for how strongly actions alter the predicted velocity at each scale.

Although the no-op behavior is structurally encouraged by centering the action at $\boldsymbol{a}_\varnothing$, real demonstrations may contain near-null commands and small stationary segments rather than exact no-op actions. When such reliable segments are

available, we add a calibration term on the action-control residual:

$$\mathcal{L}_{\text{noop}} = \mathbb{E}_{t\in\mathcal{N}} \sum_{s=1}^{S} \|\Delta \boldsymbol{v}^s_{\text{ctrl}}(\cdot, \boldsymbol{a}_t)\|_2^2, \tag{18}$$

where $\mathcal{N}$ is the set of stationary or near-null-action segments. For datasets without reliable null-action segments, we set $\lambda_{\text{noop}} = 0$.

### *E. Pyramid-Consistent Coordinates & Rollout-Aware Training*

Continuous latent autoregression is susceptible to accumulated error: small deviations in early predictions become part of the history for later predictions. SAMPO++ uses two complementary mechanisms to reduce this effect. PC-RoPE gives the latent pyramid a consistent coordinate system across scales, and rollout-aware training exposes the model to its own generated histories during training.

**Pyramid-Consistent RoPE.** A naive scale-aware RoPE can be obtained by adding a scale index [52] to the usual time and spatial rotations. This does not account for the fact that one grid step at a $1 \times 1$ scale and one grid step at a $16 \times 16$ scale represent different physical displacements. PC-RoPE instead places every token on a resolution-normalized coordinate frame while also retaining an explicit scale rotation. For a token at scale $s$ and spatial index $(h, w)$, the normalized coordinate is

$$\boldsymbol{p}_{s,h,w} = \left(\frac{h+0.5}{H_s}, \frac{w+0.5}{W_s}\right) \in [0,1)^2. \tag{19}$$

The scale-specific frequency band is defined as:

$$\boldsymbol{\Omega}_s = \boldsymbol{M}_s \boldsymbol{\Omega}, \tag{20}$$

where $\boldsymbol{\Omega}$ is the base rotary frequency bank and $\boldsymbol{M}_s$ is a scale-dependent frequency mask. Coarse scales keep low-frequency channels, while finer scales progressively open higher-frequency channels. For attention, query and key tokens may come from different coordinates. Let the query coordinate be $(t_q, s_q, h_q, w_q)$ and the key coordinate be $(t_k, s_k, h_k, w_k)$. PC-RoPE applies

$$\begin{aligned} \boldsymbol{q}' &= \boldsymbol{q}\boldsymbol{R}_t(t_q)\boldsymbol{R}_s(s_q)\boldsymbol{R}_p^{\boldsymbol{\Omega}_{s_q}}(\boldsymbol{p}_{s_q,h_q,w_q}), \\ \boldsymbol{k}' &= \boldsymbol{k}\boldsymbol{R}_t(t_k)\boldsymbol{R}_s(s_k)\boldsymbol{R}_p^{\boldsymbol{\Omega}_{s_k}}(\boldsymbol{p}_{s_k,h_k,w_k}), \end{aligned} \tag{21}$$

where $\boldsymbol{R}_t$, $\boldsymbol{R}_s$, and $\boldsymbol{R}_p^{\boldsymbol{\Omega}_s}$ are the temporal, scale, and spatial rotary transforms, respectively. PC-RoPE therefore reuses the standard rotary mechanism; the change is the coordinate and frequency assignment, not the rotation itself. This coordinate assignment reduces the ambiguity between scale identity and physical displacement, which is especially important when coarse states provide structural priors for fine-scale flow rendering.

**Rollout-aware training.** The main source of long-horizon drift is the mismatch between teacher-forced training and autoregressive inference [53]–[56]. During training, the model usually conditions on ground-truth histories; during inference, it conditions on previously generated latents. To reduce this mismatch, we use scheduled self-forcing. Let $\hat{\boldsymbol{z}}_r^s$ denote a model-generated latent at video index $r$ and $\boldsymbol{z}_r^s$ the corresponding ground-truth latent. When constructing the history for a later prediction, we replace part of the ground-truth history by stopped-gradient model predictions:

$$\bar{\boldsymbol{z}}_r^s = m_r \boldsymbol{z}_r^s + (1-m_r)\text{sg}(\hat{\boldsymbol{z}}_r^s), \quad m_r \sim \text{Bernoulli}(1-p_{\text{roll}}), \tag{22}$$

where $p_{\text{roll}}$ is annealed during training. Rollout-aware batches are sampled with a fixed probability, and the truncated horizon $K$ is gradually increased, which limits the additional cost relative to teacher-forced training. The resulting truncated rollout loss is

$$\mathcal{L}_{\text{roll}} = \sum_{k=1}^{K}\sum_{s=1}^{S} \omega_{k,s} \left\|\hat{\boldsymbol{z}}_{t+k}^s - \text{sg}(\boldsymbol{z}_{t+k}^s)\right\|_2^2, \tag{23}$$

where $K$ is the truncated rollout horizon, and $\omega_{k,s}$ controls the loss weight at horizon $k$ and scale $s$. Stop-gradient is applied only when generated latents are reused as historical context; the current rollout prediction still receives gradients through $\mathcal{L}_{\text{roll}}$. In practice, the rollout loss is applied more strongly

**Algorithm 1** Action-Controlled Velocity Field (ACVF)

**Require:** $\boldsymbol{u}$: hidden feature, $\boldsymbol{c}_{t+1,s}$: action-free condition, $\boldsymbol{h}_{t+1}^s$: scale state, $\boldsymbol{a}_t$: action, $\boldsymbol{a}_\varnothing$: no-op action, $s$: scale index

$\boldsymbol{u}_n \leftarrow \text{RMSNorm}(\boldsymbol{u})$ *# Normalize feature*
$\boldsymbol{v}_{\text{drift}} \leftarrow f_{\text{drift}}^s(\boldsymbol{u}_n; \boldsymbol{c}_{t+1,s})$ *# Passive drift*
$\bar{\boldsymbol{a}}_t \leftarrow \boldsymbol{a}_t - \boldsymbol{a}_\varnothing$ *# Center action*
$\boldsymbol{b}_{t,s} \leftarrow \boldsymbol{B}_s \psi(\bar{\boldsymbol{a}}_t)$ *# Scale action basis*
$\boldsymbol{\phi}_{t,s} \leftarrow \boldsymbol{W}_o(\boldsymbol{u}_n \odot \boldsymbol{b}_{t,s})$ *# Control atom*
$\boldsymbol{g}_{t+1,s} \leftarrow \gamma_{\max} \tanh(\boldsymbol{W}_g \boldsymbol{h}_{t+1}^s)$ *# Bounded gain*
$\boldsymbol{m}_{t+1,s} \leftarrow \sigma(\text{MLP}([\boldsymbol{u}_n, \boldsymbol{h}_{t+1}^s]))$ *# Spatial mask*
$\Delta\boldsymbol{v}_{\text{ctrl}} \leftarrow \eta_s \boldsymbol{m}_{t+1,s} \odot (\boldsymbol{g}_{t+1,s} \odot \boldsymbol{\phi}_{t,s})$ *# Action residual*
$\boldsymbol{v}_\theta^s \leftarrow \boldsymbol{v}_{\text{drift}} + \Delta\boldsymbol{v}_{\text{ctrl}}$ *# Compose velocity*
**return** $\boldsymbol{v}_\theta^s$

**Algorithm 2** SAMPO++ Action-Conditioned Rollout

**Require:** $\boldsymbol{o}_{1:T_c}$: context observations, $\boldsymbol{a}_{T_c:T_c+H-1}$: action plan, $\mathcal{E}$: encoder, $\mathcal{D}$: decoder, $N$: ODE steps

$\mathcal{H}_z \leftarrow \{\mathcal{E}(\boldsymbol{o}_\tau)^{1:S}\}_{\tau=1}^{T_c}$ *# Encode context*
$\mathcal{H}_a \leftarrow \{\boldsymbol{a}_{1:T_c-1}\}$ *# Initialize actions*
**for** $k = 0, \ldots, H-1$ **do**
  $t \leftarrow T_c + k \quad \boldsymbol{a}_t \leftarrow \boldsymbol{a}_{T_c+k}$ *# Current action*
  $\{\boldsymbol{h}_{t+1}^s\}_{s=1}^S \leftarrow \Phi_\theta^{\text{time}}(\mathcal{H}_z, \mathcal{H}_a)$ *# Predict scale states*
  $\boldsymbol{z}_{t+1}^{<1} \leftarrow \emptyset$ *# Start pyramid*
  **for** $s = 1, \ldots, S$ **do**
    $\boldsymbol{x}_0^s \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ *# Sample prior*
    $v_\theta^s \leftarrow \text{ACVF}(\boldsymbol{x}_\tau^s, \tau; \boldsymbol{z}_{t+1}^{<s}, \boldsymbol{h}_{t+1}^s, \boldsymbol{a}_t)$ *# Build velocity*
    $\boldsymbol{z}_{t+1}^s \leftarrow \text{ODESolve}(\boldsymbol{x}_0^s, v_\theta^s, N)$ *# Integrate flow*
    $\boldsymbol{z}_{t+1}^{<s+1} \leftarrow \boldsymbol{z}_{t+1}^{<s} \cup \{\boldsymbol{z}_{t+1}^s\}$ *# Update pyramid*
  **end for**
  $\mathcal{H}_z \leftarrow \mathcal{H}_z \cup \{\boldsymbol{z}_{t+1}^{1:S}\} \quad \mathcal{H}_a \leftarrow \mathcal{H}_a \cup \{\boldsymbol{a}_t\}$ *# Update history*
**end for**
$\hat{\boldsymbol{o}}_{T_c+1:T_c+H} \leftarrow \mathcal{D}(\boldsymbol{z}_{T_c+1:T_c+H}^{1:S})$ *# Decode rollout*
**return** $\hat{\boldsymbol{o}}_{T_c+1:T_c+H}$

to coarse and middle scales, while fine-scale losses remain dominated by the flow objective.

**Optional optical-flow rectification.** For comparison with inference-time guidance methods, we retain optical-flow rectification [57] as an optional diagnostic baseline. It steers the sampling velocity toward a warped previous latent using an externally estimated motion prior. We do not use it as the main long-horizon mechanism because it requires a motion field for a frame that is still being generated and relies on pixel-space assumptions that may not align with latent dynamics. Our main results therefore focus on rollout-aware training, with rectification reported only as a controlled comparison.

**Overall objective.** The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{Flow}} + \lambda_{\text{plan}}\mathcal{L}_{\text{plan}} + \lambda_{\text{cf}}\mathcal{L}_{\text{cf}} + \lambda_{\text{noop}}\mathcal{L}_{\text{noop}} + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}}, \tag{24}$$

where $\lambda_{\text{plan}}$, $\lambda_{\text{cf}}$, $\lambda_{\text{noop}}$, and $\lambda_{\text{roll}}$ are scalar loss weights. We set $\lambda_{\text{noop}} = 0$ for datasets without reliable null-action or stationary segments. Optional optical-flow rectification is disabled unless explicitly stated. This separation keeps the central training signals focused on flow matching, scale-wise planning, action-discriminative control, no-op calibration when available, and rollout consistency. Algorithm 1 summarizes the ACVF computation, and Algorithm 2 summarizes action-conditioned rollout.

### F. SAMPO++ as an Action-Controlled Neural Simulator

Beyond open-loop future prediction, the same transition model can be used as a learned action-conditioned simulator for closed-loop imagined execution. The goal is not to select among multiple candidate policies at every step. Instead, a policy is executed inside SAMPO++ over the full horizon: the policy outputs one action chunk, the world model rolls out the visual consequence of that chunk, and the last predicted observation is fed back to the policy to produce the next chunk. This chunk-level feedback loop matches how many VLA policies are deployed [58]–[60], where a short action segment is predicted from the current observation and the policy is queried again after the segment is executed.

Fig. 3 illustrates this simulator use case. Let $L$ denote the action-chunk length and let $K$ denote the number of chunks, so that the imagined horizon contains $H = KL$ low-level action steps. We initialize the imagined trajectory with the real observation, namely $\hat{\boldsymbol{o}}_t = \boldsymbol{o}_t$ and $\hat{\boldsymbol{z}}_t = \mathcal{E}(\boldsymbol{o}_t)$. For chunk index $n = 0, \ldots, K-1$, the VLA policy is queried on the latest simulated observation and language instruction $\ell$ to produce the next action chunk:

$$\hat{\boldsymbol{a}}_{t+nL:t+(n+1)L-1} \sim \pi\left(\cdot \mid \hat{\boldsymbol{o}}_{t+nL}, \ell\right), \tag{25}$$

where $\hat{\boldsymbol{a}}_{t+nL:t+(n+1)L-1}$ denotes the entire action segment executed during the $n$-th imagined chunk. SAMPO++ then applies its next-state transition model step by step inside the chunk:

$$\begin{aligned} \hat{\boldsymbol{z}}^{1:S}_{t+nL+j+1} &= \text{Rollout}_\theta\left(\hat{\boldsymbol{z}}^{1:S}_{\le t+nL+j}, \hat{\boldsymbol{a}}_{t+nL+j}\right), \\ \hat{\boldsymbol{o}}_{t+nL+j+1} &= \mathcal{D}\left(\hat{\boldsymbol{z}}^{1:S}_{t+nL+j+1}\right), \quad j = 0, \ldots, L-1, \end{aligned} \tag{26}$$

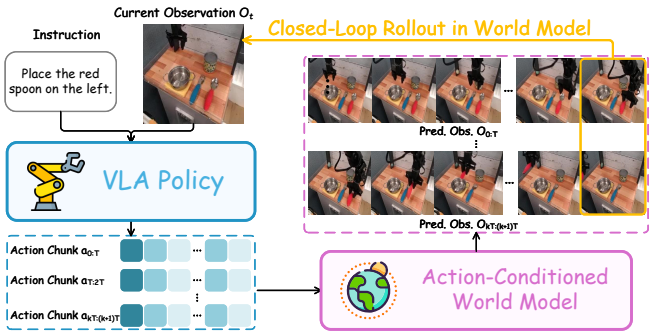


Fig. 3: **Closed-loop imagined execution in SAMPO++.** A VLA policy predicts an action chunk, SAMPO++ rolls out the corresponding future observations, and the last predicted frame is fed back to the policy for the next chunk.

where hatted variables denote imagined quantities generated by the simulator, and $\text{Rollout}_\theta$ denotes one application of the transition model in Eq. (5). After the chunk is completed, the last predicted observation $\hat{\boldsymbol{o}}_{t+(n+1)L}$ becomes the new policy input for the next chunk. Thus, errors in the generated visual state can affect subsequent actions, making this setting a closed-loop test of both the policy and the world model rather than a teacher-forced video prediction task.

When task-level evaluation is desired, the imagined execution can be assigned a trajectory score [2], [46], [61] after the rollout is completed:

$$J_\theta(\pi) = \mathbb{E}\left[\sum_{k=1}^{H} \gamma^{k-1} R_\psi\left(\hat{\boldsymbol{z}}^{1:S}_{t+k}, \ell\right)\right], \tag{27}$$

where $\gamma$ is a discount factor and $R_\psi$ is a learned reward head when dense rewards are available. For success-only tasks, $R_\psi$ can be replaced by a success/progress estimator $S_\psi$ or by a task-specific progress function. These heads are trained only when reward, success, or task-progress supervision is available and are not required for visual prediction. If a downstream policy also requires proprioception, a separate proprioceptive transition head $Q_\chi(\hat{\boldsymbol{z}}^{1:S}_{t+1}, \boldsymbol{h}^{1:S}_{t+1}, \boldsymbol{a}_t)$ predicts the next proprioceptive state $\hat{\boldsymbol{q}}_{t+1}$ when such supervision is available.

This simulator view does not claim to replace real-environment evaluation. Its role is to provide a closed-loop imagined execution setting in which an entire action sequence is generated and executed inside the learned world model chunk by chunk. This use case directly motivates our action-centric diagnostics: the model must remain action-sensitive within each chunk, counterfactually discriminative across different actions, and stable when the last predicted observation is repeatedly fed back to the policy.

## IV. Experiments and Analysis

### A. Experimental Setup

**Experimental goal.** Our experiments are organized to test whether SAMPO++ is merely a stronger video predictor or an *action-controlled world model* (ACWM). We therefore evaluate four complementary properties. First, the renderer should preserve visual realism on standard video prediction

TABLE I: **Video prediction results on BAIR and RoboNet datasets.** Following standard evaluation protocols, models are tasked with predicting 15 future frames conditioned on a single context frame for the BAIR ($1 \rightarrow 15$ frames), and 10 future frames conditioned on two context frames for the RoboNet ($2 \rightarrow 10$ frames). We evaluate the generation fidelity on both datasets across 3 random seeds and report the mean and standard deviation. "-" indicates values not reported in the original papers. Note that LPIPS and SSIM are scaled by 100 for better readability.

| **BAIR** [24] | FVD↓ | PSNR↑ | SSIM↑ | LPIPS↓ | **RoboNet** [25] | FVD↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|---|---|---|---|---|
| *action-free & 64×64 resolution* | | | | | *action-conditioned & 64×64 resolution* | | | | |
| FitVid [62] | 93.6 | - | - | - | FitVid [62] | 62.5 | 28.2 | 89.3 | **2.4** |
| MCVD [63] | 89.5 | 16.9 | 78.0 | - | SVG [64] | 123.2 | 23.9 | 87.8 | 6.0 |
| MAGVIT [14] | 62.0 | 19.3 | 78.7 | 12.3 | GHVAE [65] | 95.2 | 24.7 | 89.1 | 3.6 |
| iVideoGPT [7] | 75.0 | 20.4 | 82.3 | 9.5 | iVideoGPT [7] | 63.2 | 27.8 | 90.6 | 4.9 |
| SAMPO [8] | 65.7 | 22.3 | 86.7 | 8.4 | SAMPO [8] | 57.1 | 29.3 | 94.1 | 3.3 |
| SAMPO++ (Ours) | **59.4**$^{\pm 1.5}$ | **23.1**$^{\pm 0.2}$ | **88.5**$^{\pm 0.4}$ | **7.2**$^{\pm 0.1}$ | SAMPO++ (Ours) | **49.8**$^{\pm 1.2}$ | **30.5**$^{\pm 0.1}$ | **95.2**$^{\pm 0.2}$ | 2.6$^{\pm 0.1}$ |
| *action-conditioned & 64×64 resolution* | | | | | *action-conditioned & 256×256 resolution* | | | | |
| iVideoGPT [7] | 60.8 | 24.5 | 90.2 | 5.0 | iVideoGPT [7] | 197.9 | 23.8 | 80.8 | 14.7 |
| SAMPO [8] | 55.5 | 26.7 | 94.7 | 3.7 | SAMPO-L [8] | 175.3 | 25.3 | 84.7 | 12.3 |
| SAMPO++ (Ours) | **47.6**$^{\pm 1.8}$ | **27.9**$^{\pm 0.3}$ | **96.1**$^{\pm 0.3}$ | **3.1**$^{\pm 0.1}$ | SAMPO++ (Ours) | **152.4**$^{\pm 3.5}$ | **26.8**$^{\pm 0.2}$ | **87.5**$^{\pm 0.5}$ | **9.8**$^{\pm 0.3}$ |

TABLE II: **Video prediction results on the 1X dataset.** We predict 10 future frames from 2 context frames.

| **1X WM** [66] | FVD↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|
| *Action-Conditioned* ($256 \times 256$) | | | | |
| iVideoGPT [7] | 251.8 | 24.1 | 78.3 | 20.3 |
| SAMPO [8] | 227.1 | 25.7 | 80.3 | 18.7 |
| SAMPO++ (Ours) | **188.5** | **26.8** | **83.1** | **16.4** |

benchmarks. Second, the transition should be controlled by the current action rather than by passive visual inertia. Third, the learned dynamics should remain stable under long-horizon autoregressive rollout. Fourth, the model should be useful when queried as an imagined simulator for planning and policy learning. Perceptual metrics are treated as a prerequisite; action alignment, counterfactual discrimination, no-op stability, rollout drift, and downstream control are the central evidence for ACWM.

**Datasets.** We evaluate SAMPO++ across robotic manipulation, embodied control, and action-free driving. **(1) Robotic manipulation.** BAIR [24] provides stochastic pushing sequences with end-effector actions and is used to test short-horizon action-conditioned dynamics. RoboNet [25] evaluates cross-embodiment and cross-viewpoint generalization over videos from multiple robotic platforms. The 1X World Model dataset [66] stresses high-resolution generation in cluttered humanoid teleoperation scenes. **(2) Embodied control and policy learning.** VP$^2$ [28] is used to evaluate whether the learned predictor can support model-predictive visual planning, and Meta-World [29] is used to test whether the world model can provide useful imagined rollouts for downstream reinforcement learning. **(3) Autonomous driving.** Cityscapes [26] and OpenDV-YouTube [27] are evaluated in an action-free future prediction setting. These driving results are not used as evidence for action controllability; instead, they test whether the same visual dynamics prior remains stable in long-horizon, high-motion outdoor scenes.

**Pre-training data.** SAMPO++ uses the same Open X-Embodiment pre-training mixture as our conference version. Concretely, the mixture contains 41 datasets and 1,407,330 trajectories from Open X-Embodiment [67]. To avoid domination by a few large datasets, we follow the same capped sampling scheme used in the conference paper: high-resource datasets are assigned a capped sampling weight, while low-resource datasets are retained with non-zero weights. This detail is important for reproducibility because the pre-training stage is intended to provide a broad robot-manipulation visual prior rather than an in-domain advantage for any single downstream benchmark. Unless otherwise stated, all SAMPO++ variants in the ablations use this identical Open X-Embodiment pre-training mixture and differ only in the components being tested.

**Baselines.** We compare SAMPO++ with representative discrete and continuous world-model baselines. In the discrete autoregressive regime, we include iVideoGPT [7] and the conference version SAMPO [8]. In the continuous generative regime, we compare with SyncVP [68] and STDiff [69]. For reproduced baselines and our variants, we match the pre-training and fine-tuning budgets whenever possible. For literature numbers that are not fully reproducible under our exact pre-training mixture, we report the published setting and avoid using them to support component-level claims. Component-level conclusions are drawn from controlled variants of the same SAMPO++ backbone.

**Metrics.** We report conventional perceptual metrics, including FVD [47], PSNR [48], SSIM [70], and LPIPS [71]. These metrics evaluate visual realism and reconstruction quality, but they do not by themselves establish whether the transition is controlled by action. We therefore additionally report action-centric and rollout-centric diagnostics. Action alignment ($AA_{cos}$) measures the cosine alignment between the predicted controllable latent displacement and the ground-truth latent transition. Counterfactual accuracy (CF Acc.) measures the percentage of cases in which the factual action explains the future better than a temporally mismatched action under the same visual context. No-op residual (No-op Res.) measures the normalized magnitude of the action-control residual under null or near-null commands, reported as a percentage.

TABLE III: **Long-horizon video prediction on BAIR and Cityscapes.** We report standard long-horizon prediction and extended prediction. BAIR evaluates action-conditioned robotic dynamics, while Cityscapes evaluates action-free outdoor visual dynamics under large ego-motion. #T denotes the number of sampled trajectories used for evaluation. Following Tab. I, SSIM and LPIPS are scaled by 100 for readability.

| **BAIR** [24] | #T | FVD↓ | SSIM↑ | LPIPS↓ | **Cityscapes** [26] | #T | FVD↓ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|---|---|---|---|---|
| *Standard Prediction: 2 → 28 frames* | | | | | *Standard Prediction: 2 → 28 frames* | | | | |
| MCVD [72] | 10 | 120.6 | 78.5 | 7.1 | VDT [73] | – | 142.3 | **88.0** | – |
| SAVP [74] | 100 | 116.4 | 78.9 | 6.3 | MCVD [72] | 10 | 141.31 | 69.0 | 11.2 |
| ExtDM-K4 [75] | 100 | 102.8 | 81.4 | 6.9 | ExtDM-K4 [75] | 100 | 121.3 | 74.5 | 10.8 |
| STDiff [69] | 10 | 88.1 | 81.8 | 6.9 | STDiff [69] | 10 | 107.31 | 65.8 | 13.6 |
| SyncVP [68] | 10 | 63.6 | 80.5 | 7.5 | SyncVP [68] | 10 | 84.00 | 64.9 | 16.0 |
| SAMPO [8] | 10 | 65.7 | 86.7 | 5.9 | SAMPO [8] | 10 | 92.50 | 76.5 | 12.5 |
| SAMPO++ (Ours) | 10 | **52.5** | **88.5** | **4.6** | SAMPO++ (Ours) | 10 | **73.45** | 78.8 | **9.5** |
| *Extended Prediction: 2 → 48 frames* | | | | | *Extended Prediction: 2 → 48 frames* | | | | |
| SyncVP [68] | 10 | 92.1 | 74.8 | 8.4 | SyncVP [68] | 10 | 165.80 | 56.5 | 19.2 |
| SAMPO [8] | 10 | 81.3 | 83.2 | 6.7 | SAMPO [8] | 10 | 145.60 | 68.0 | 15.6 |
| SAMPO++ (Ours) | 10 | **64.8** | **86.1** | **5.4** | SAMPO++ (Ours) | 10 | **118.42** | **71.5** | **13.5** |

TABLE IV: **Quantitative evaluation of visual planning efficacy on the control-centric VP$^2$ benchmark.** We report the downstream Task Success Rate (%) across eight distinct robotic manipulation tasks. Following standard protocol [7], [8], the aggregated average performance (*Avg. Success*) explicitly excludes the "Flat Block" task due to its inherently low simulator upper bound. All metrics represent the mean and standard deviation computed across 3 independent random seeds.

| Method / Task | Robosuite Push | Flat Block | Open Drawer | Open Slide | Blue Button | Green Button | Red Button | Upright Block | Avg. Success↑ |
|---|---|---|---|---|---|---|---|---|---|
| Simulator | $93.5^{\pm 1.2}$ | $13.3^{\pm 0.0}$ | $76.7^{\pm 0.0}$ | $71.7^{\pm 1.4}$ | $100.0^{\pm 0.0}$ | $96.7^{\pm 0.0}$ | $90.0^{\pm 0.0}$ | $90.0^{\pm 0.0}$ | 100.0 |
| FitVid [62] | $67.7^{\pm 6.4}$ | $\mathbf{9.2}^{\pm 2.8}$ | $25.3^{\pm 8.2}$ | $35.3^{\pm 5.5}$ | $94.0^{\pm 4.2}$ | $84.0^{\pm 5.5}$ | $58.7^{\pm 5.5}$ | $51.3^{\pm 2.9}$ | 65.6 |
| SVG' [64] | $79.8^{\pm 3.3}$ | $2.0^{\pm 1.4}$ | $16.7^{\pm 8.2}$ | $\mathbf{57.3}^{\pm 11.0}$ | $97.3^{\pm 2.7}$ | $81.3^{\pm 5.5}$ | $76.0^{\pm 9.5}$ | $48.7^{\pm 11.1}$ | 72.5 |
| MCVD [63] | $77.3^{\pm 2.1}$ | $4.0^{\pm 1.4}$ | $11.7^{\pm 1.4}$ | $18.3^{\pm 1.4}$ | $95.0^{\pm 4.1}$ | $83.3^{\pm 0.0}$ | $73.3^{\pm 2.7}$ | $56.7^{\pm 2.7}$ | 64.3 |
| Struct-VRNN [76] | $55.4^{\pm 4.1}$ | $4.7^{\pm 5.5}$ | $2.7^{\pm 4.2}$ | $12.7^{\pm 6.9}$ | $86.7^{\pm 4.2}$ | $68.0^{\pm 9.5}$ | $30.7^{\pm 4.2}$ | $33.3^{\pm 2.7}$ | 44.2 |
| iVideoGPT [7] | $78.3^{\pm 0.8}$ | $3.3^{\pm 0.7}$ | $37.5^{\pm 1.7}$ | $16.1^{\pm 2.7}$ | $95.6^{\pm 2.1}$ | $82.5^{\pm 3.4}$ | $92.2^{\pm 1.4}$ | $44.7^{\pm 1.7}$ | 70.1 |
| SAMPO [8] | $80.7^{\pm 1.4}$ | $5.5^{\pm 1.2}$ | $40.3^{\pm 2.3}$ | $18.3^{\pm 3.3}$ | $97.3^{\pm 1.7}$ | $85.3^{\pm 3.3}$ | $94.7^{\pm 2.1}$ | $46.1^{\pm 2.7}$ | 72.2 |
| SAMPO++ (Ours) | $\mathbf{86.2}^{\pm 1.1}$ | $6.8^{\pm 0.9}$ | $\mathbf{48.5}^{\pm 1.5}$ | $32.4^{\pm 2.1}$ | $\mathbf{98.5}^{\pm 0.5}$ | $\mathbf{91.5}^{\pm 1.2}$ | $\mathbf{96.0}^{\pm 1.1}$ | $\mathbf{64.5}^{\pm 2.4}$ | **75.4** |

$\text{Drift}_{100}$ measures accumulated latent deviation during a 100-step self-conditioned rollout. For downstream utility, we report VP$^2$ MPC success, policy-level agreement under closed-loop imagined execution when available, and Meta-World success curves. Quantitative results are averaged over multiple seeds; standard deviations are reported for action-centric diagnostics when repeated runs are available.

**Implementation details.** SAMPO++ uses the continuous pyramidal tokenizer and the scale-wise flow renderer described in Sec. III. The tokenizer and the temporal-transition backbone are initialized from Open X-Embodiment pre-training, then fine-tuned on each downstream benchmark. During controlled ablations, the tokenizer, model size, action representation, optimizer, fine-tuning budget, and evaluation protocol are kept fixed; only the tested component is changed. We train on 8 NVIDIA A800 GPUs using AdamW with a base learning rate of $1 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, a 5k-step linear warmup followed by cosine decay, weight decay 0.01, and `bfloat16` mixed precision.

### *B. High-Fidelity Video Prediction*

**Short-term prediction.** We first evaluate whether the continuous scale-wise renderer preserves visual quality before asking whether the dynamics are action controlled. Tab. I reports BAIR and RoboNet results, and Tab. II evaluates high-resolution humanoid manipulation. These results should be interpreted as a visual-fidelity check rather than direct evidence of ACWM: a model can look visually plausible while still ignoring the action. As shown in Tab. I, SAMPO++ remains competitive across perceptual metrics. On RoboNet at $256 \times 256$, it attains an FVD of **152.4**, improving over the discrete baseline SAMPO and iVideoGPT. We attribute the gain to continuous scale-wise flow matching, which avoids the high-frequency flickering and texture loss often introduced by codebook quantization. On the 1X World Model dataset, SAMPO++ also improves FVD and LPIPS over iVideoGPT. The key point is not that continuous latents alone solve world modeling, but that they provide a cleaner state space for the action-controlled dynamics introduced in Sec. III-D.

**Long-term generation.** Long-horizon autoregressive prediction is a stronger stress test because errors compound over generated frames. Tab. III evaluates both action-conditioned robotic rollouts on BAIR and action-free outdoor rollouts on Cityscapes under standard and extended horizons. On BAIR, SAMPO++ improves the $2 \rightarrow 28$ and $2 \rightarrow 48$ FVD scores over prior diffusion and autoregressive baselines, indicating that the multi-scale planner and rollout-aware training reduce accumulated drift when generated frames are fed back into

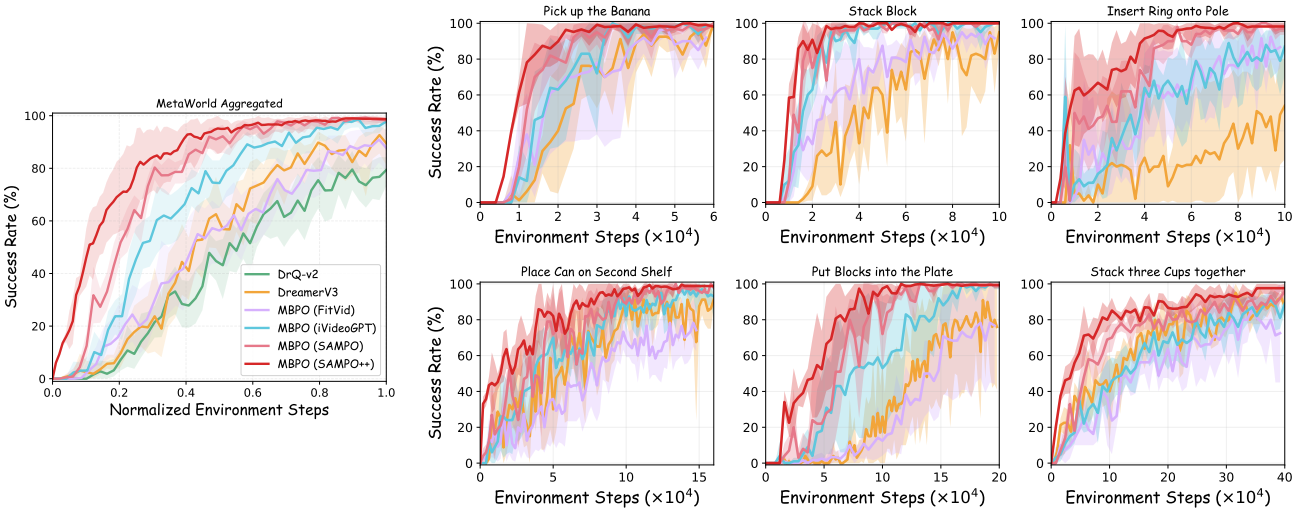


Fig. 4: **Model-based RL on Meta-World with SAMPO++. Left:** Aggregated performance reports the Interquartile Mean (IQM) and 95% confidence intervals over 30 runs (5 seeds × 6 tasks). **Right:** Task-specific learning curves show mean success rates and 95% confidence intervals across 5 random seeds. Success is evaluated over 20 independent episodes per checkpoint.

the model. On Cityscapes, SAMPO++ also maintains stronger long-horizon visual dynamics under ego-motion, which shows that the learned visual dynamics prior remains stable even without action supervision.

**Generalization to autonomous driving.** We next examine action-free long-horizon video generation on autonomous-driving data. Cityscapes and OpenDV-YouTube introduce large camera motion, complex lighting, and moving agents, but they do not provide the same action-supervised control signal as robotic manipulation datasets. We therefore use these benchmarks to test the scalability of the visual dynamics prior rather than to claim action controllability. The Cityscapes results are reported in the right half of Tab. III; SAMPO++ improves long-term FVD and LPIPS under both horizons, complementing the robotic experiments with an action-free stress test of visual dynamics. Additional OpenDV-YouTube videos and quantitative results are shown on the project page to avoid diluting the main ACWM evidence chain.

### C. Embodied Control & Policy Learning

**Visual planning on VP$^2$.** VP$^2$ is designed as a control-centric benchmark [28] for video prediction: the predictor is queried through a single forward-prediction interface inside a model-predictive controller. We follow the official VP$^2$ evaluation code and task definitions, using the benchmark's planning interface and reporting MPC success rates. This setup is stricter than open-loop video metrics because a visually plausible but action-insensitive predictor cannot reliably choose action sequences that accomplish the task. Tab. IV shows that SAMPO++ improves average planning success, indicating that the gains in action alignment and rollout stability translate into downstream control.

**Model-based RL on Meta-World.** We also evaluate whether SAMPO++ can support model-based policy learning. Following the iVideoGPT visual MBRL protocol [7], we use Meta-World tasks with visual observations and train a DrQ-v2-style actor-critic [77] while augmenting the replay buffer with imagined rollouts from the learned world model. This follows the MBPO-style decoupling [78] of model learning and policy learning: the world model generates short synthetic interaction segments, and the actor-critic is trained from a mixture of real and imagined experience. We use the same task set and evaluation convention as the iVideoGPT protocol: success is measured in the ground-truth Meta-World simulator over held-out evaluation episodes, while the model is used only to provide additional imagined rollouts during policy optimization. For sparse or unstable task rewards, we use the standard success-bonus shaping used in the prior protocol:

$$r = r_{\text{task}} + 10.0 \cdot \mathbb{I}_{\text{task success}}. \tag{28}$$

This experiment is an application-level utility check rather than the primary evidence for ACWM; the primary evidence remains the action-centric metrics and the stepwise construction study.

Fig. 4 shows that SAMPO++ improves both aggregate sample efficiency and task-wise learning stability. This does not mean that SAMPO++ is an RL algorithm; rather, it indicates that a more action-sensitive and rollout-stable world model produces synthetic transitions that are less misleading for downstream actor-critic learning. The Meta-World experiment therefore complements the VP$^2$ planning results: VP$^2$ tests model-predictive action selection with a fixed planner, whereas Meta-World tests whether imagined rollouts remain useful when repeatedly consumed by a learning policy.

### D. Action-Controlled World Modeling

Perceptual prediction quality is necessary but insufficient for embodied world modeling. A model can produce visually plausible futures while still ignoring the action, collapsing different controls to similar outcomes, or drifting when its own predictions are fed back into the policy. We therefore evaluate SAMPO++ with a world-model-native protocol that

TABLE V: **Stepwise construction of SAMPO++ as an action-controlled world model.** Starting from a continuous scale-wise renderer, we progressively add history modeling, explicit action control, action-level objectives, pyramid-consistent coordinates, and rollout-aware training. AA is reported as cosine similarity, while CF Acc. and No-op Res. are reported as percentages. Action-centric metrics are reported as mean with standard deviation over three random seeds.

| Configuration | History | Action Control | Coordinates | Rollout Training | Objective | $\text{FVD}_{48}\downarrow$ | $\text{AA}_{\cos}\uparrow$ | CF Acc. (%)↑ | No-op Res. (%)↓ | $\text{Drift}_{100}\downarrow$ | $\text{VP}^2\uparrow$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Continuous renderer | single | passive cond. | legacy | teacher-forced | – | 92.5 | $0.50^{\pm 0.01}$ | $71.0^{\pm 2.0}$ | $20.0^{\pm 1.0}$ | 0.74 | 72.0 |
| + temporal planner | multi | passive cond. | legacy | teacher-forced | – | 84.0 | $0.57^{\pm 0.02}$ | $75.0^{\pm 1.0}$ | $17.0^{\pm 1.0}$ | 0.62 | 73.2 |
| + ACVF | multi | ACVF | legacy | teacher-forced | – | 78.5 | $0.71^{\pm 0.02}$ | $87.0^{\pm 2.0}$ | $7.0^{\pm 1.0}$ | 0.55 | 74.2 |
| + no-op calibration | multi | ACVF | legacy | teacher-forced | $\mathcal{L}_{\text{noop}}$ | 76.8 | $0.73^{\pm 0.01}$ | $88.0^{\pm 1.0}$ | $3.0^{\pm 0.6}$ | 0.53 | 74.5 |
| + counterfactual action | multi | ACVF | legacy | teacher-forced | $\mathcal{L}_{\text{noop}} + \mathcal{L}_{\text{cf}}$ | 74.2 | $0.78^{\pm 0.02}$ | $93.0^{\pm 1.0}$ | $3.0^{\pm 0.5}$ | 0.50 | 74.9 |
| + PC-RoPE | multi | ACVF | PC-RoPE | teacher-forced | $\mathcal{L}_{\text{noop}} + \mathcal{L}_{\text{cf}}$ | 69.0 | $0.79^{\pm 0.01}$ | $93.0^{\pm 1.0}$ | $3.0^{\pm 0.5}$ | 0.39 | 75.1 |
| **SAMPO++** | **multi** | **ACVF** | **PC-RoPE** | **rollout-aware** | $\mathcal{L}_{\text{noop}} + \mathcal{L}_{\text{cf}} + \mathcal{L}_{\text{roll}}$ | **64.8** | $\mathbf{0.81}^{\pm 0.01}$ | $\mathbf{94.0}^{\pm 0.8}$ | $\mathbf{2.0}^{\pm 0.4}$ | **0.28** | **75.4** |

TABLE VI: **Closed-loop policy execution inside SAMPO++.** Success rates are averaged over four SIMPLER WidowX Bridge tasks with 200 evaluation episodes per task.

| Policy | SAMPO++ SR ↑ | SIMPLER SR ↑ | $\lvert\Delta\rvert\downarrow$ |
|---|---|---|---|
| $\pi_0$ [60] | 62.0 | 66.0 | 4.0 |
| OpenVLA [59] | 47.5 | 45.0 | 2.5 |
| Octo [58] | 38.5 | 41.0 | 2.5 |

directly tests whether the latent dynamics are action-sensitive, discriminative under action replacement, stable under null commands, and robust under long-horizon self-conditioned rollout.

**World-model-native metrics.** All action-centric diagnostics are computed in the latent space used by the model before RGB decoding. Let $\Delta\hat{\boldsymbol{z}}_{t,s}(\boldsymbol{a})$ denote the predicted pooled latent displacement at scale $s$ under action $\boldsymbol{a}$, and let $\Delta\boldsymbol{z}^{\star}_{t,s}$ denote the corresponding ground-truth pooled displacement. Action alignment measures whether the predicted transition follows the factual transition direction:

$$\text{AA}_{\cos} = \mathbb{E}_{t,s}\left[\cos\left(\Delta\hat{\boldsymbol{z}}_{t,s}(\boldsymbol{a}_t), \Delta\boldsymbol{z}^{\star}_{t,s}\right)\right]. \tag{29}$$

Counterfactual accuracy measures whether the factual action predicts the target transition better than a mismatched action under the same history, planner state, and sampling condition. We first define the prediction errors under the factual and mismatched actions as:

$$D_t^{+} = d(\hat{\boldsymbol{z}}_{t+1}(\boldsymbol{a}_t), \boldsymbol{z}_{t+1}), \quad D_t^{-} = d(\hat{\boldsymbol{z}}_{t+1}(\tilde{\boldsymbol{a}}_t), \boldsymbol{z}_{t+1}). \tag{30}$$

The counterfactual accuracy is then

$$\text{CF Acc.} = 100 \cdot \mathbb{E}_t\left[\mathbf{1}(D_t^{+} < D_t^{-})\right]. \tag{31}$$

where $d(\cdot,\cdot)$ is the latent distance used for evaluation and $\tilde{\boldsymbol{a}}_t$ is sampled from temporally mismatched action chunks. No-op residual measures whether the action-control branch remains quiet under null or near-null commands:

$$\text{Noop Res.} = 100 \cdot \mathbb{E}_{t\in\mathcal{N},s}\left[\frac{\|\Delta\boldsymbol{v}^{s}_{\text{ctrl}}(\cdot,\boldsymbol{a}_t)\|_2}{\|\boldsymbol{v}^{s}_{\text{drift}}\|_2 + \epsilon}\right]. \tag{32}$$

$\text{Drift}_{100}$ measures accumulated latent deviation over a 100-step self-conditioned rollout:

$$\text{Drift}_{100} = \frac{1}{100}\sum_{k=1}^{100}\left\|\hat{\boldsymbol{z}}^{1:S}_{t+k} - \boldsymbol{z}^{1:S}_{t+k}\right\|_2. \tag{33}$$

Thus, $\text{AA}_{\cos}$ and CF Acc. are better when higher, while No-op Res. and $\text{Drift}_{100}$ are better when lower. These metrics complement FVD and LPIPS: FVD measures perceptual realism, whereas $\text{AA}_{\cos}$, CF Acc., No-op Res., and $\text{Drift}_{100}$ test whether the generated dynamics are action-sensitive, discriminative under action replacement, stable under null commands, and robust under self-conditioned rollout.

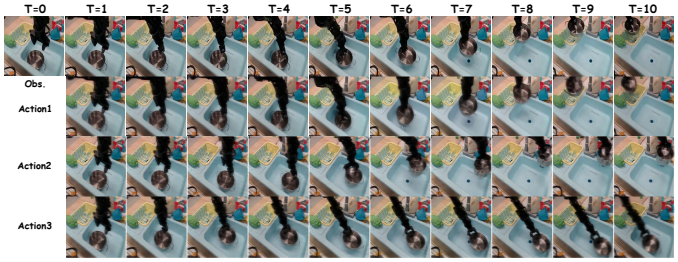


Fig. 5: **Counterfactual control visualization.** The first row shows the ground-truth future trajectory. The remaining three rows show imagined rollouts generated by SAMPO++ under the same initial observation but 3 different action chunks.

Tab. V summarizes these diagnostics under a stepwise construction of SAMPO++. Each row adds one mechanism from Sec. III, allowing the table to be read as a property-level test rather than a generic module ablation. Starting from a continuous scale-wise renderer, the multi-scale temporal planner reduces $\text{Drift}_{100}$ by maintaining separate history states for different spatial scales. ACVF substantially improves $\text{AA}_{\cos}$ and CF Acc. by routing the current action through an explicit residual velocity path. No-op calibration directly suppresses unnecessary action-control residuals under null or near-null commands, while the counterfactual action objective further improves discrimination between factual and mismatched actions. PC-RoPE mainly improves long-horizon consistency by separating physical position, scale identity, and rotary frequency. Finally, rollout-aware training replaces the teacher-forced training interface with a self-conditioned one, reducing rollout drift and improving downstream $\text{VP}^2$ success. The final row improves not only FVD but also $\text{AA}_{\cos}$, CF Acc., No-op Res., $\text{Drift}_{100}$, and $\text{VP}^2$ average success, which is the key evidence that SAMPO++ moves beyond perceptual prediction toward action-controlled world modeling.

**Counterfactual control visualization.** To make the action-conditioned behavior of SAMPO++ visually interpretable, we compare imagined futures generated from the same initial observation history under different action chunks. As shown in Fig. 5, the first row presents the ground-truth future trajectory, while the remaining three rows show the corresponding

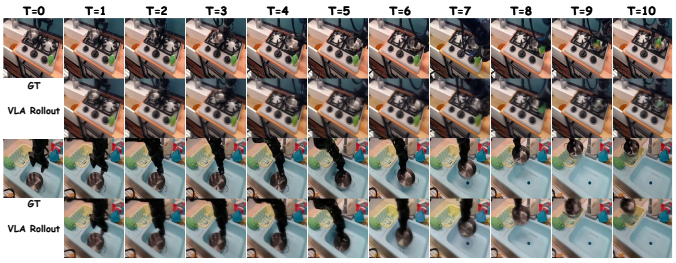


Fig. 6: **Closed-loop policy rollout inside SAMPO++.** For each example, the first row shows the ground-truth trajectory and the second row shows the imagined rollout generated by executing the VLA policy inside SAMPO++.

rollouts predicted by SAMPO++ when only the action chunk is changed. The observation history and model state are fixed across the three imagined branches, so the differences across predicted futures reflect sensitivity to the action input rather than stochastic variation in the context. SAMPO++ produces clearly distinct future evolutions for different action chunks, indicating that the learned dynamics remain sensitive to executed controls rather than collapsing to a single visually plausible future. This qualitative evidence is consistent with the $AA_{cos}$, CF Acc., No-op Res., and $Drift_{100}$ results reported in Tab. V.

**Closed-loop policy rollout inside ACWM.** We next evaluate the simulator use case described in Sec. III-F. We use four SIMPLER WidowX Bridge tasks: `widowx carrot on plate`, `widowx put eggplant in basket`, `widowx spoon on towel`, and `widowx stack cube`. For each VLA policy and each task, we run 200 evaluation episodes in both SAMPO++ and the ground-truth SIMPLER [79] environment, using the same task instructions, initial-state distribution, rollout horizon, and success predicate.

Inside SAMPO++, the policy is executed in a chunk-level closed loop. Given the current simulated observation, the policy predicts an action chunk; SAMPO++ rolls out the corresponding future observations; the last predicted frame is then fed back to the same policy to produce the next action chunk. This differs from open-loop video prediction because visual errors can change later policy inputs and compound into action errors. Therefore, policy rollout inside SAMPO++ is a stricter test of ACWM than per-frame prediction: the model must remain action-sensitive within each chunk and stable when its own predicted frames become future policy inputs.

Tab. VI compares the success rates obtained by executing the same VLA policies inside SAMPO++ and in the ground-truth SIMPLER environment. This experiment is not intended to replace real-environment evaluation; instead, it tests whether the learned world model preserves the relative outcomes of different policies under closed-loop imagined execution. SAMPO++ preserves the policy ordering observed in SIMPLER, yielding a Spearman rank correlation of $\rho = 1.00$ over the evaluated policies. The absolute success-rate errors are small, with a mean error of 3.0 points and a maximum error of 4.0 points across the three policies. These results suggest that SAMPO++ has potential as a lightweight proxy for preliminary policy screening before expensive environment

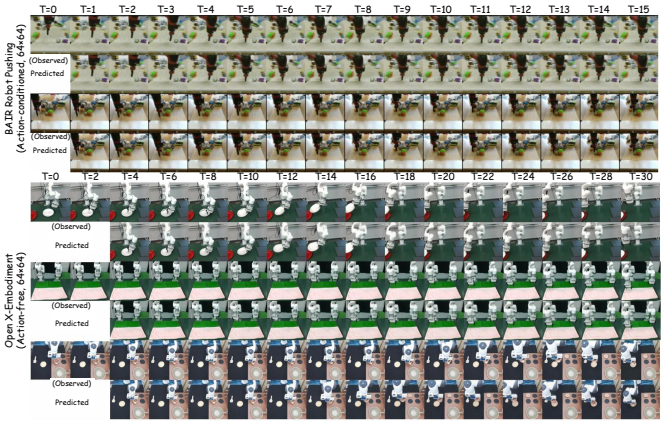


Fig. 7: **Visualization on BAIR and Open X-Embodiment datasets.** Please zoom in to examine the visual details.

rollouts. Importantly, this policy-level agreement depends on the same ACWM properties evaluated above: if the model ignores actions, fails under null commands, or drifts under self-conditioning, closed-loop policy outcomes quickly become unreliable.

**Closed-loop rollout visualization.** Fig. 6 compares ground-truth trajectories with closed-loop imagined rollouts produced by executing a VLA policy inside SAMPO++. For each example, the first row is the ground-truth trajectory and the second row is the corresponding SAMPO++ rollout. Unlike Fig. 5, which studies counterfactual action effects under a fixed history, this figure emphasizes repeated policy–world-model interaction: the policy consumes the latest simulated frame, predicts an action chunk, and the last predicted frame is fed back as the next policy input. This setting tests whether SAMPO++ can maintain action-conditioned dynamics over multiple closed-loop interaction cycles.

### *E. Qualitative Analysis & Visualization*

To contextualize the quantitative results, we present qualitative rollouts across robotic manipulation, visual planning, policy rollout, and driving. The paper includes representative cases, while longer videos, additional tasks, and failure cases are provided on the project page.[1] These visualizations are not intended to replace the quantitative metrics; instead, they illustrate the behaviors measured by our evaluation protocol, including action sensitivity, object permanence, long-horizon drift, and closed-loop feedback stability.

**Robotic manipulation across resolutions.** Fig. 7 shows BAIR and Open X-Embodiment rollouts, demonstrating that SAMPO++ preserves object motion and gripper-object interactions at standard resolution. Fig. 9 shows high-resolution RoboNet and 1X rollouts, where the model preserves fine texture and object permanence in more cluttered scenes. These visualizations support the perceptual results in Tab. I and Tab. II, but they should be read together with the action-centric metrics in Tab. V.

**Action-conditioned rollouts for control.** Fig. 10 visualizes action-conditioned predictions on VP$^2$. Because the same action sequence is provided to the predictor and the ground-truth

[1] https://sanmumumu.github.io/SAMPO_plus_plus/

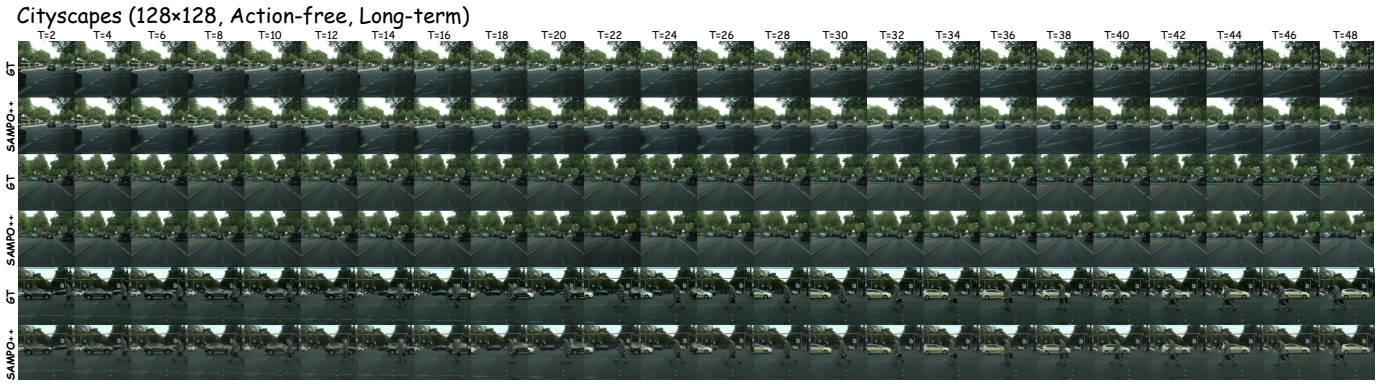


Fig. 8: **Long-term video prediction on the Cityscapes dataset.** We visualize three driving scenarios. SAMPO++ maintains plausible scene dynamics and background structure over long rollouts despite ego-motion and large pixel displacement. These action-free videos are used to evaluate visual-dynamics stability rather than action controllability.

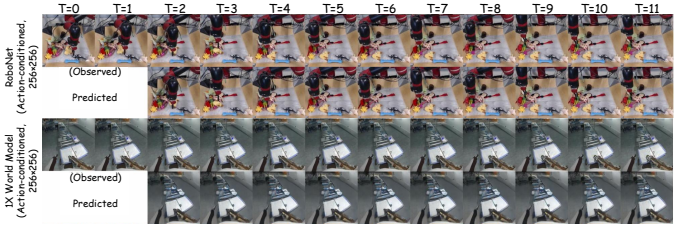


Fig. 9: **Visualization on RoboNet and 1X World Model.**

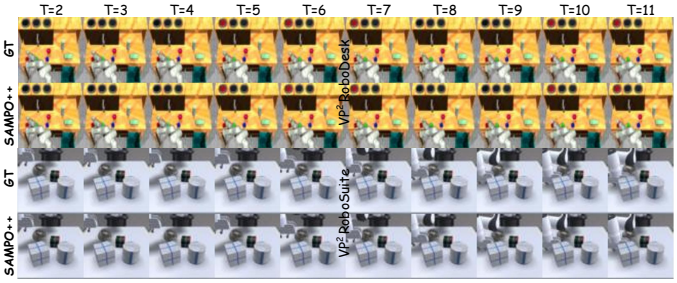


Fig. 10: **Visualization on the VP$^2$ benchmark.** We show representative RoboDesk and RoboSuite manipulation tasks, where predicted rollouts are conditioned on the same action sequences used by the planner.

simulator, discrepancies reflect model errors rather than planner differences. The qualitative results show that SAMPO++ can preserve action-conditioned object motion over multiple steps, which explains why the predictor can be used inside MPC.

**Action-free driving and project-page visualizations.** Fig. 8 visualizes Cityscapes long-horizon rollouts. These videos demonstrate large-motion visual dynamics but are not used as evidence for action controllability. Additional OpenDV videos, no-op examples, counterfactual branches, Meta-World rollouts, and failure cases are placed on the project page to keep the main paper focused on the ACWM evidence chain.

## V. Conclusion and Discussion

In this work, we revisit action-conditioned world modeling from the perspective of controlled dynamics. Rather than treating a world model as a conventional action-conditioned video generator, we formulate it as a scale-decoupled transition model whose predictions should remain sensitive to actions and stable under autoregressive rollout. Based on this principle, we present SAMPO++, a continuous-latent framework that couples temporal autoregression with scale-wise flow matching. A multi-scale temporal planner summarizes the latent history with scale-matched recurrent states, an Action-Controlled Velocity Field decomposes the generative velocity into passive drift and bounded action-induced residual dynamics, and Pyramid-Consistent RoPE aligns tokens across time, scale, and normalized spatial coordinates. Together with rollout-aware training, these components aim to improve both action controllability and long-horizon consistency. Experiments across robotic manipulation, visual planning, model-based reinforcement learning, and action-free driving video prediction show that SAMPO++ improves perceptual prediction quality while providing stronger evidence of world-model utility through action-centric diagnostics. In particular, action alignment, counterfactual accuracy, no-op stability, rollout drift, and downstream control performance provide a more direct assessment of whether the learned model behaves as an action-controlled transition operator, rather than merely producing visually plausible futures. Ablation studies further show that the proposed planner, ACVF, PC-RoPE, and rollout-aware training contribute to different aspects of controllability and stability.

## References

[1] D. Ha and J. Schmidhuber, "World models," *NeurIPS*, 2018.
[2] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *ICLR*, 2020.
[3] Y. LeCun, "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27," *Open Review*, vol. 62, no. 1, pp. 1–62, 2022.
[4] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," *arXiv preprint arXiv:2010.02193*, 2020.
[5] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.
[6] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps *et al.*, "Genie: Generative interactive environments," in *ICML*, 2024.
[7] J. Wu, S. Yin, N. Feng, X. He, D. Li, J. Hao, and M. Long, "ivideogpt: Interactive videogpts are scalable world models," in *NeurIPS*, 2024.
[8] S. Wang, J. Tian, L. Wang, Z. Liao, J. Li, H. Dong, K. Xia, S. Zhou, W. Tang, and H. Gang, "Sampo: Scale-wise autoregression with motion prompt for generative world models," *NeurIPS*, 2025.
[9] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *NeurIPS*, vol. 30, 2017.
[10] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *CVPR*, 2021, pp. 12 873–12 883.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[12] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021.

[13] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, "Latent video transformer," *arXiv preprint arXiv:2006.10704*, 2020.

[14] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa *et al.*, "Magvit: Masked generative video transformer," in *CVPR*, 2023, pp. 10 459–10 469.

[15] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, V. Birodkar, A. Gupta, X. Gu *et al.*, "Language model beats diffusion–tokenizer is key to visual generation," *ICLR*, 2024.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[17] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022, pp. 10 684–10 695.

[18] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *ICLR*, 2023.

[19] X. Liu, C. Gong *et al.*, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *ICLR*, 2023.

[20] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *ICCV*, 2023, pp. 4195–4205.

[21] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023.

[22] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *ICCV*, 2023, pp. 3836–3847.

[23] C. Mou, X. Wang, L. Xie, Y. Wu, J. Zhang, Z. Qi, and Y. Shan, "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," in *AAAI*, vol. 38, no. 5, 2024, pp. 4296–4304.

[24] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections." *CoRL*, vol. 12, no. 16, p. 23, 2017.

[25] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *CoRL*, 2019.

[26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, June 2016.

[27] J. Yang, S. Gao, Y. Qiu, L. Chen, T. Li, B. Dai, K. Chitta, P. Wu, J. Zeng, P. Luo, J. Zhang, A. Geiger, Y. Qiao, and H. Li, "Generalized predictive model for autonomous driving," in *CVPR*, 2024.

[28] S. Tian, C. Finn, and J. Wu, "A control-centric benchmark for video prediction," *ICLR*, 2023.

[29] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *CoRL*. PMLR, 2020, pp. 1094–1100.

[30] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang, "Visual autoregressive modeling: Scalable image generation via next-scale prediction," *NeurIPS*, vol. 37, pp. 84 839–84 865, 2024.

[31] T. Li, Y. Tian, H. Li, M. Deng, and K. He, "Autoregressive image generation without vector quantization," *NeurIPS*, vol. 37, pp. 56 424–56 445, 2024.

[32] S. Ren, Q. Yu, J. He, X. Shen, A. Yuille, and L.-C. Chen, "Flowar: Scale-wise autoregressive image generation meets flow matching," *ICML*, 2025.

[33] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *ICLR*, 2021.

[34] Y. Liu, K. Zhang, Y. Li, Z. Yan, C. Gao, R. Chen, Z. Yuan, Y. Huang, H. Sun, J. Gao *et al.*, "Sora: A review on background, technology, limitations, and opportunities of large vision models," *arXiv preprint arXiv:2402.17177*, 2024.

[35] M. Albergo, N. M. Boffi, and E. Vanden-Eijnden, "Stochastic interpolants: A unifying framework for flows and diffusions," *Journal of Machine Learning Research*, vol. 26, no. 209, pp. 1–80, 2025.

[36] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie, "Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers," in *ECCV*. Springer, 2024, pp. 23–40.

[37] S. Wang, L. Wang, S. Zhou, J. Tian, J. Li, H. Sun, and W. Tang, "Flowram: Grounding flow matching policy with region-aware mamba framework for robotic manipulation," in *CVPR*, 2025, pp. 12 176–12 186.

[38] C. Zhou, L. Yu, A. Babu, K. Tirumala, M. Yasunaga, L. Shamis, J. Kahn, X. Ma, L. Zettlemoyer, and O. Levy, "Transfusion: Predict the next token and diffuse images with one multi-modal model," *ICLR*, 2025.

[39] J. Xie, W. Mao, Z. Bai, D. J. Zhang, W. Wang, K. Q. Lin, Y. Gu, Z. Chen, Z. Yang, and M. Z. Shou, "Show-o: One single transformer to unify multimodal understanding and generation," *ICLR*, 2025.

[40] J. Parker-Holder, P. Ball, J. Bruce, V. Dasagi, K. Holsheimer, C. Kaplanis, A. Moufarek, G. Scully, J. Shar, J. Shi, S. Spencer, J. Yung, M. Dennis, S. Kenjeyev, S. Long, V. Mnih, H. Chan, M. Gazeau, B. Li, F. Pardo, L. Wang, L. Zhang, F. Besse, T. Harley, A. Mitenkova, J. Wang, J. Clune, D. Hassabis, R. Hadsell, A. Bolton, S. Singh, and T. Rocktäschel, "Genie 2: A large-scale foundation world model," 2024. [Online]. Available: https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/

[41] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter, "Diffusion models are real-time game engines," *ICLR*, 2025.

[42] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *CVPR*, 2023, pp. 15 619–15 629.

[43] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus *et al.*, "V-jepa 2: Self-supervised video models enable understanding, prediction and planning," *arXiv preprint arXiv:2506.09985*, 2025.

[44] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "Unisim: A neural closed-loop sensor simulator," in *CVPR*, 2023, pp. 1389–1399.

[45] Q. Dao, H. Phung, B. Nguyen, and A. Tran, "Flow matching in latent space," *arXiv preprint arXiv:2307.08698*, 2023.

[46] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *NeurIPS*, vol. 32, 2019.

[47] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018.

[48] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[49] A. Davtyan, S. Sameni, and P. Favaro, "Efficient video prediction via sparsely conditioned flow matching," in *ICCV*, 2023, pp. 23 263–23 274.

[50] D. Geng, C. Herrmann, J. Hur, F. Cole, S. Zhang, T. Pfaff, T. Lopez-Guevara, C. Doersch, Y. Aytar, M. Rubinstein *et al.*, "Motion prompting: Controlling video generation with motion trajectories," *CVPR*, 2025.

[51] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel, "Learning interactive real-world simulators," *ICLR*, vol. 1, no. 2, p. 6, 2023.

[52] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[53] X. Huang, Z. Li, G. He, M. Zhou, and E. Shechtman, "Self forcing: Bridging the train-test gap in autoregressive video diffusion," *NeurIPS*, 2025.

[54] B. Chen, D. Martí Monsó, Y. Du, M. Simchowitz, R. Tedrake, and V. Sitzmann, "Diffusion forcing: Next-token prediction meets full-sequence diffusion," *NeurIPS*, vol. 37, pp. 24 081–24 125, 2024.

[55] D. Zhou, Q. Sun, Y. Peng, K. Yan, R. Dong, D. Wang, Z. Ge, N. Duan, X. Zhang, L. M. Ni *et al.*, "Taming teacher forcing for masked autoregressive video generation," *CVPR*, 2025.

[56] M. Pasini, J. Nistal, S. Lattner, and G. Fazekas, "Continuous autoregressive models with noise augmentation avoid error accumulation," *arXiv preprint arXiv:2411.18447*, 2024.

[57] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *ECCV*. Springer, 2020, pp. 402–419.

[58] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[59] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[60] K. Black *et al.*, "$\pi_0$: A vision-language-action flow model for general robot control," *Robotics: Science and Systems XXI*, 2024.

[61] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[62] M. Babaeizadeh, M. T. Saffar, S. Nair, S. Levine, C. Finn, and D. Erhan, "Fitvid: Overfitting in pixel-level video prediction," *arXiv preprint arXiv:2106.13195*, 2021.
[63] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, "Mcvd-masked conditional video diffusion for prediction, generation, and interpolation," *NeurIPS*, vol. 35, pp. 23 371–23 385, 2022.
[64] R. Villegas, A. Pathak, H. Kannan, D. Erhan, Q. V. Le, and H. Lee, "High fidelity video prediction with large stochastic recurrent neural networks," *NeurIPS*, vol. 32, 2019.
[65] B. Wu, S. Nair, R. Martin-Martin, L. Fei-Fei, and C. Finn, "Greedy hierarchical variational autoencoders for large-scale video prediction," in *CVPR*, 2021, pp. 2318–2328.
[66] 1X Technologies, "1X World Model Challenge," Jun. 2024. [Online]. Available: https://www.1x.tech/discover/1x-world-model
[67] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *ICRA*. IEEE, 2024, pp. 6892–6903.
[68] E. Pallotta, S. M. Azar, S. Li, O. Zatsarynna, and J. Gall, "Syncvp: Joint diffusion for synchronous multi-modal video prediction," in *CVPR*, June 2025, pp. 13 787–13 797.
[69] X. Ye and G.-A. Bilodeau, "Stdiff: Spatio-temporal diffusion for continuous stochastic video prediction," in *AAAI*, 2024.
[70] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004.
[71] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.
[72] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, "Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation," in *NeurIPS*, 2022. [Online]. Available: https://arxiv.org/abs/2205.09853
[73] H. Lu, G. Yang, N. Fei, Y. Huo, Z. Lu, P. Luo, and M. Ding, "Vdt: General-purpose video diffusion transformers via mask modeling," in *ICLR*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258832473
[74] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv preprint arXiv:1804.01523*, 2018.
[75] Z. Zhang, J. Hu, W. Cheng, D. Paudel, and J. Yang, "Extdm: Distribution extrapolation diffusion model for video prediction," in *CVPR*, 2024.
[76] M. Minderer, C. Sun, R. Villegas, F. Cole, K. P. Murphy, and H. Lee, "Unsupervised learning of object structure and dynamics from videos," *NeurIPS*, vol. 32, 2019.
[77] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Mastering visual continuous control: Improved data-augmented reinforcement learning," *ICLR*, 2022.
[78] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," in *NeurIPS*, 2019.
[79] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani *et al.*, "Evaluating real-world robot manipulation policies in simulation," *CoRL*, 2024.

**Sen Wang** received the B.E. degree in Control Science and Engineering from Jilin University, Changchun, China, in 2023. He is currently a third-year Ph.D. candidate in the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include embodied computer vision, robotic manipulation and interactive world model.

**Sanping Zhou** (Member, IEEE) received the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2020. From 2018 to 2019, he was a visiting Ph.D. student with Robotics Institute, Carnegie Mellon University. He is currently a professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include machine learning, deep learning, computer vision, and embodied AI, with a focus on person re-identification, salient object detection, medical image segmentation, image classification, and visual tracking.

**Huaiyi Dong** received the B.E. degree in Software Engineering from China University of Geosciences, wuhan, China, in 2023. He is currently a Ph.D. candidate in the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include embodied computer vision and robotic manipulation.

**Kun Xia** received the Ph.D. degree in Control Science and Engineering from Xi'an Jiaotong University, Xi'an, China, in 2024. From 2022 to 2023, he was a visiting Ph.D. student with University of Illinois Chicago, IL, USA. He is currently an Assistant Professor with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China. His research interests include computer vision, image/video processing, analysis and understanding.

**Gang Hua** (Fellow, IEEE) received the B.S. and M.S. degrees in automatic control engineering from Xi'an Jiaotong University, Xi'an, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering and computer science from Northwestern University, Evanston, IL, USA, in 2006. He was a senior scientist with Microsoft Live labs Research from 2006 to 2009, a senior researcher with Nokia Research Center Hollywood from 2009 to 2010, a research staff member with IBM Research T. J. Watson Center from 2010 to 2011, and a visiting researcher from 2011 to 2014. During 2014–15, he took a leave and worked on the Amazon-Go project. He was an associate professor with the Stevens Institute of Technology from 2011 to 2015. He was also with Microsoft from 2015 to 2018 as the Science/Technical adviser to the CVP of the Computer Vision Group, director of Computer Vision Science Team in Redmond, Taipei ATL, and senior principal researcher/research manager with Microsoft Research. He was the CTO with Convenience Bee, and the managing director and chief scientist of its research branch in US, Wormpex AI Research, from 2018 to 2024. He was the vice president with Multimodal Experiences Research Lab, Dolby Laboratories from 2024 to 2025. He is currently the director of applied science with Amazon Alexa AI. His research interests include computer vision, pattern recognition, machine learning, robotics, towards general artificial intelligence, with primary applications in cloud and edge intelligence. He is an associate editor for TPAMI and MVA. He is a general chair of ICCV'2027 and a program chair of CVPR'2019 & 2022. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award. He is an IAPR Fellow and an ACM distinguished scientist.

**Le Wang** (Senior Member, IEEE) received the B.S. and Ph.D. degrees in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2008 and 2014, respectively. From 2013 to 2014, he was a visiting Ph.D. student with the Stevens Institute of Technology, Hoboken, NJ, USA. From 2016 to 2017, he was a visiting scholar with Northwestern University, Evanston, IL, USA. He is currently a professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, machine learning, and robotic manipulation. He is an associate editor for PR, MVA, and PRL.