LaMo: A Latent Motion World Model for Long-Horizon Prediction

Anonymous authors
Paper under double-blind review

Abstract

Visual world models learned from video are often trained to predict future observations from past frames, but pixel-space forecasting is expensive and forces the model to allocate capacity to high-frequency details. Latent-space video models reduce this cost, yet they typically predict a dense latent state at every step, repeatedly carrying largely static information and making long-horizon rollouts harder to sustain. We propose LaMo, a probabilistic latent dynamics world model that predicts *motion residuals*—continuous latent variables encoding differences between consecutive motion tokens in a learned latent dynamics space. Given the current latent state and a short history of past motion tokens, our model autoregressively samples the next motion residual, integrates it with the previous motion token to obtain the next motion token, and recurrently advances the latent state using a frozen forward dynamics model. We instantiate the same contextual transformer backbone with two probabilistic parameterizations for residual prediction: Gaussian mixtures and conditional flow matching. We evaluate long-horizon rollouts on BDD100K driving scenes and on two robot-manipulation datasets, BridgeData V2 and RoboCasa, using image-space metrics and, on BDD100K, semantic segmentation mIoU. Motion-residual prediction consistently improves long-horizon rollouts over an architecture-matched dense latent-state baseline while requiring $3.8\times$ fewer FLOPs per predicted frame.

1 Introduction

Predictive world models enable embodied agents to *imagine* potential trajectories, enabling planning and control. For these agents to be useful over long horizons, world model rollouts must remain temporally coherent rather than drifting or hallucinating (Janner et al., 2019). Long-horizon prediction is therefore central to video-based world models for embodied decision-making (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; Yang et al., 2023; Jang et al., 2025). Recent generative models operate in latent space to mitigate the cost of pixel-level forecasting (Blattmann et al., 2023a; Bar-Tal et al., 2024; Ma et al., 2025; Assran et al., 2025). However, these latent spaces are typically reconstructed via autoencoders that retain low-level details and result in dense token representations per frame. This can introduce significant redundancy as videos are typically dominated by static content, which can saturate model context and increase computational cost. Consequently, the temporal consistency and generation quality of these generative models can degrade over longer horizons (Xie et al., 2024; Cai et al., 2025; Yang et al., 2025). Recent long-video systems address this issue with specialized mechanisms such as progressive autoregressive denoising or sparse long-context routing (Xie et al., 2024; Cai et al., 2025). These systems are complementary to our goal: rather than competing with frontier video generators trained with much larger data, compute, and specialized long-context architectures, we compare motion-residual prediction against dense latent-state prediction under a matched transformer backbone and training setup.

To tackle the long-horizon prediction problem, we propose to capitalize on the efficiency advantages of modeling *latent dynamics* over latent states. Intuitively, dynamics is often of lower complexity than appearance; e.g., a rigid object undergoes coherent motion governed by a small number of shared degrees of freedom, so isolating dynamics can offer substantial compression benefits. Therefore, we aim to model videos through latent *motion tokens*, compact variables that represent transition dynamics in a learned latent space. Latent motion tokens, also known as latent actions, stem from a recent line of work (Bruce et al., 2024; Schmidt & Jiang, 2024; Ye et al., 2025; Cui et al., 2024)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

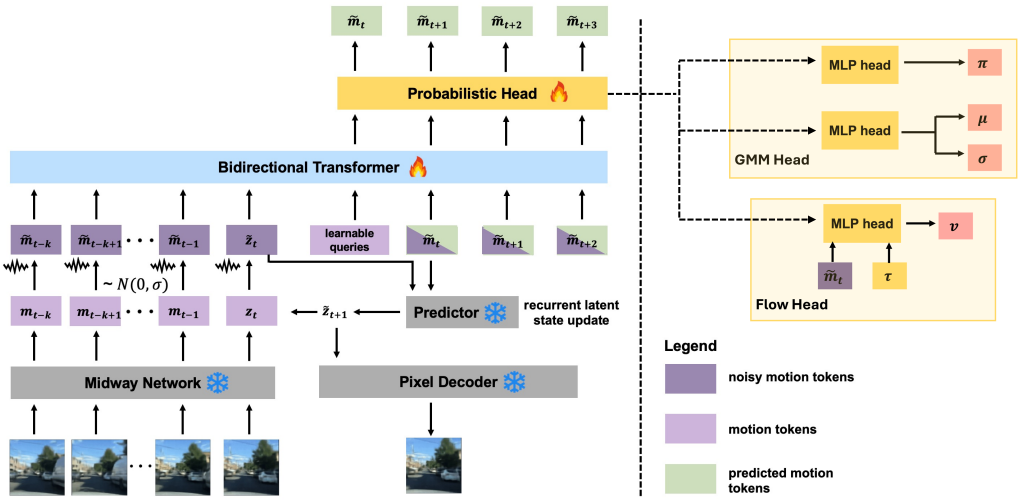


Figure 1: **LaMo overview.** A frozen latent dynamics model encodes frames into latent states z_t and extracts motion tokens m_t. LaMo predicts motion residuals Δm_t with either a GMM or flow-matching head. At inference, predicted residuals are integrated into motion tokens, which update the latent state through the frozen forward dynamics model; decoded latents yield pixel-space rollouts.

that seeks to learn compressed representations of transition dynamics and models that can predict future observations conditioned on these latent representations. We specifically adopt Midway Network (Hoang & Ren, 2026), which employs a hierarchical formulation that refines latent motion tokens across multiple feature levels to handle complex scene dynamics in natural videos. However, Midway Network and other latent motion methods are restricted to inference between observed frames or only one-step prediction. To bridge this gap, we propose to learn a conditional generative model that predicts how latent motion changes over long horizons given a short history of motion tokens and the current latent state.

We introduce LaMo, a probabilistic world model designed for long-horizon prediction in latent motion space. Rather than predicting the next motion token directly, LaMo predicts a *motion residual*: the difference between consecutive motion tokens. This residual formulation shifts the prediction problem from modeling absolute transition dynamics to modeling how the dynamics themselves change over time. Given a predicted residual, we reconstruct the next motion token by adding it to the previous motion token and then use the frozen forward dynamics predictor to advance the latent state. A transformer backbone processes the context inputs and produces residual-aligned features. A probabilistic head then generates the next motion residual, using either **(a)** a Gaussian mixture model (GMM) trained by conditional likelihood or **(b)** a flow matching head that learns a conditional velocity field. To improve robustness under autoregressive rollouts, we apply context-noise augmentation to the conditioning motion token history and current latent state (Pasini et al., 2024; Valevski et al., 2024). The reconstructed motion tokens recurrently update the latent state through the frozen forward dynamics predictor, and the resulting latent trajectory can be decoded for visualization or used for downstream tasks.

In our experiments on BDD100K (Yu et al., 2020), a large-scale driving dataset, LaMo produces more accurate long-horizon rollouts than an autoregressive transformer that conditions on and predicts dense latent states, while using $3.8\times$ fewer FLOPs per predicted frame. The same advantage holds on two robot-manipulation datasets, BridgeData V2 (Walke et al., 2023) and RoboCasa (Nasiriany et al., 2024), where residual prediction improves FID, PSNR, and SSIM at every horizon. We also observe that the two probabilistic head variants offer distinct benefits: the flow-matching head achieves lower FID across time horizons, while the GMM head attains better PSNR and SSIM in long-horizon settings.

To summarize, our contributions are:

- We propose LaMo, a probabilistic latent dynamics world model for long-horizon prediction that models videos through compact *motion residuals*, i.e., differences between consecutive latent motion tokens.

- We show that predicting motion residuals and integrating them into motion tokens yields more accurate and stable long-horizon rollouts than a state-prediction baseline that conditions on dense latent state history, across driving and robot-manipulation datasets and at substantially lower computational cost.
- We evaluate rollouts with image-space metrics and semantic segmentation mIoU, showing that motion-residual prediction better preserves scene-level structure.

2 Related Work

Future prediction. Neuroscience proposes that intelligent agents predict future sensory inputs to perceive and act, establishing predictive coding (Srinivasan et al., 1982; Rao & Ballard, 1999) and forward models (Wolpert et al., 1995; Miall & Wolpert, 1996) as central mechanisms in biological vision and motor control. In deep learning, this has motivated a long line of works in video prediction. Early attempts leveraged LSTM (Srivastava et al., 2015), CNN (Mathieu et al., 2016), and motion-focused (Finn et al., 2016) architectures and showed that trained models could be useful for generation, action recognition, and robotic control. Modern video generators such as Sora (Brooks et al., 2024), Veo (Google DeepMind, 2025), and Wan (Wan et al., 2025) combine transformer architectures (Dosovitskiy et al., 2021; Peebles & Xie, 2022), diffusion modeling (Ho et al., 2020; Gupta et al., 2024), and massive scaling to generate realistic videos. However, consistent long-horizon prediction remains challenging due to computational cost and context limits. Prior works use spatiotemporal tokenization (Yan et al., 2021; Villegas et al., 2022), latent diffusion (Blattmann et al., 2023b;a), progressive autoregressive denoising (Xie et al., 2024), or sparse long-context routing (Cai et al., 2025). In contrast, our controlled study asks whether predicting in a compact latent motion space can reduce redundancy: instead of repeatedly predicting dense latent states, we predict residual changes in compact latent motion.

Latent dynamics. Deep learning has aimed to acquire dynamics models that emulate how animals predict their environment's future (Ha & Schmidhuber, 2018). While initial works used supervised learning with action-labeled data (Agrawal et al., 2016; Pathak et al., 2017), recent efforts learn latent dynamics from unlabeled videos using discretization or dimensional bottlenecks to prevent representation collapse (Garrido et al., 2026). These latent dynamics models can support data labeling (Schmidt & Jiang, 2024; Jang et al., 2025; Ye et al., 2025), environment simulation (Bruce et al., 2024; Gao et al., 2025), or motion-prior pretraining (Chen et al., 2024b). Our two-stage setup follows a common pattern in this line of work: first learn a latent action or motion representation, then train a downstream dynamics model, policy, or autoregressive predictor on top of that representation (Bruce et al., 2024; Schmidt & Jiang, 2024; Ye et al., 2025; Chen et al., 2024b). In particular, our work leverages Midway Network (Hoang & Ren, 2026), a latent dynamics architecture that uses hierarchical refinement and dense forward prediction to infer motion tokens in natural videos. While prior works either infer latent actions post-hoc or predict one step in the future, we learn to predict residual changes in latent motion over long horizons and use the reconstructed motion tokens to evolve latent states.

Forecasting in feature space. DINO-Foresight (Karypidis et al., 2025a) and DINO-WM (Zhou et al., 2025) forecast complete future feature maps of frozen vision foundation models (VFMs), the dense formulation our state-prediction baseline follows. Closest to our motivation, DeltaTok (Kerssies et al., 2026) reduces temporal redundancy by encoding the difference between consecutive VFM feature maps as a single continuous delta token. LaMo instead predicts residuals between hierarchical motion tokens, $\Delta m_t = m_t - m_{t-1}$, and integrates them through a learned forward dynamics model. Unlike these methods, whose frozen general-purpose encoders transfer across domains, including zero-shot, LaMo relies on a Midway backbone trained per domain; building motion-residual prediction on frozen VFM features is a natural extension.

3 Background

Motion tokens. Motion tokens are learned latent vectors designed to capture transition dynamics between source and target states z_t and z_{t+1}. They are typically inferred by an inverse dynamics model, $m_t = f_{\text{inv}}(z_t, z_{t+1})$, and used by a forward dynamics model to predict future states, $\hat{z}_{t+1} = f_{\text{fwd}}(z_t, m_t)$. To prevent representation collapse, where m_t trivially compresses z_{t+1}, prior methods

use bottlenecks such as VQ-VAE discretization (Schmidt & Jiang, 2024; Bruce et al., 2024; Ye et al., 2025) or dimensionality reduction (Cui et al., 2024; Gao et al., 2025; Hoang & Ren, 2026).

Midway Network (Hoang & Ren, 2026) extends this paradigm hierarchically. Motivated by coarse-to-fine optical flow refinement (Sun et al., 2018; Jonschkowski et al., 2020), it iteratively refines motion tokens in a top-down manner. For feature levels $\{l^1, l^2, ..., l^k\}$, motion tokens m^l are computed as an accumulation of higher-level motion tokens plus a learned residual:

$$m^l = m^{l+1} + f_{\text{inv}}(\hat{z}^{l+1}_{t+1}, z^l_{t+1}). \tag{1}$$

The inverse model is conditioned on the higher-level forward prediction $\hat{z}^{l+1}_{t+1}$ rather than source features, forcing the network to learn residual dynamics that correct errors from the previous scale.

Conditional flow matching. Flow-based generative models (Chen et al., 2018; Grathwohl et al., 2019) learn a time-dependent velocity field that transports samples from a simple base distribution to the data distribution via an ODE,

$$\frac{dx(\tau)}{d\tau} = v_\theta(x(\tau), \tau \mid c)\,, \tag{2}$$

where c denotes conditioning information. A common training approach is *flow matching* (Lipman et al., 2023), which regresses v_θ to the vector field induced by a probability path between base and data samples. Using the straight-line path (Liu et al., 2022),

$$x_\tau = (1-\tau)x_0 + \tau x_1, \qquad x_0 \sim \mathcal{N}(0, I),\ x_1 \sim p(\cdot \mid c), \tag{3}$$

the target velocity is $x_1 - x_0$, giving

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{x_0, x_1, \tau}\Big[\big\|v_\theta(x_\tau, \tau \mid c) - (x_1 - x_0)\big\|_2^2\Big]. \tag{4}$$

At inference, one samples $x(0) \sim \mathcal{N}(0, I)$ and integrates the ODE forward to obtain $x(1)$.

Continuous autoregressive modeling. Recent work has highlighted that sequence models can operate on *continuous-valued tokens* and attach a lightweight generative head that models each token distribution given contextual features (Li et al., 2024; Chen et al., 2024a). This decouples *inter-token dependency modeling*, handled by the backbone, from *per-token distribution modeling*, handled by a conditional generator, motivating mixtures and flow-based objectives for continuous token prediction.

4 LaMo: A Latent Motion World Model

LaMo performs long-horizon prediction by forecasting *motion residuals*—compact latent variables that describe how motion tokens change between consecutive transitions—instead of repeatedly predicting dense per-frame latent states. At each rollout step, LaMo conditions on the current latent state and motion-token history, samples the next motion residual, integrates it with the previous motion token, and advances the latent state with a fixed forward predictor. Repeating this procedure yields latent trajectories that can be decoded to pixels. Figure 1 summarizes the pipeline.

4.1 Setup: latent states, motion tokens, and motion residuals

We factor long-horizon video prediction into (i) a pretrained latent dynamics model and (ii) a learnable stochastic sequence model over residual changes in its motion token representation. In our experiments, the latent model is instantiated by Midway Network (Hoang & Ren, 2026), which provides an encoder for latent states z_t, an inverse model for motion tokens, and a forward predictor that advances the state. LaMo learns a probabilistic residual world model on top of this representation.

Formally, the inverse dynamics model defines motion tokens as

$$m_t = f_{\text{inv}}(z_t, z_{t+1}), \tag{5}$$

and the forward transition updates the state via

$$z_{t+1} = f_{\text{fwd}}(z_t, m_t). \tag{6}$$

We keep f_{inv} and f_{fwd} fixed. This separates learning a latent dynamics space from long-horizon prediction within that space, matching staged training strategies used in recent latent-action and latent-motion approaches (Bruce et al., 2024; Ye et al., 2025; Chen et al., 2024b). Rather than directly modeling m_t, we model its residual change:

$$\Delta m_t = m_t - m_{t-1}. \tag{7}$$

The learned world model predicts

$$p_\theta(\Delta m_t \mid m_{t-K:t-1},\, z_t)\,. \tag{8}$$

At inference, the predicted residual is integrated with the previous motion token,

$$\hat{m}_t = \hat{m}_{t-1} + \widehat{\Delta m}_t, \tag{9}$$

and passed to the frozen forward dynamics predictor.

4.2 Autoregressive backbone

We use a bidirectional transformer encoder (Vaswani et al., 2017) over the fixed-length context window to produce *per-motion token features*, one feature vector for each motion token slot, which parameterize a conditional distribution over the next motion residual. We embed the conditioning inputs into a common model dimension and add factorized embeddings for time index, feature level, spatial patch index when applicable, and token role.

Learnable query tokens. To produce a context representation aligned with the motion token layout, we append learnable query tokens arranged as a $(L \times N_m)$ grid. After self-attention, we take the transformer outputs at these query positions and reshape them into

$$h_t \in \mathbb{R}^{L \times N_m \times d}, \tag{10}$$

providing one contextual feature vector per motion token position. These queries are trained end-to-end through the flow or GMM prediction loss.

Global readout query (GMM only). For the GMM formulation, we additionally append a single learnable *global readout query*. Its output, $h_t^{\text{g}} \in \mathbb{R}^d$, summarizes the context and predicts the mixture weights.

4.3 Context-noise augmentation

Autoregressive rollouts induce a train–test mismatch: at inference, the model conditions on its own predicted motion tokens and latents. To improve robustness, we perturb the *conditioning inputs* during training (Pasini et al., 2024; Valevski et al., 2024):

$$\tilde{m}_{t-K:t-1} = m_{t-K:t-1} + \sigma\,\epsilon_m, \qquad \tilde{z}_t = z_t + \sigma\,\epsilon_z, \tag{11}$$

with $\epsilon_m, \epsilon_z \sim \mathcal{N}(0, I)$. For the state-prediction baseline, we analogously corrupt the latent token history $\tilde{z}_{t-K:t}$. The noise scale $\sigma \sim \mathcal{U}(0, 1)$ is drawn per sample and is not provided to the model.

4.4 Two probabilistic heads for motion residual prediction

Using the same transformer-encoded context features, we study two conditional density parameterizations for Δm_t: a GMM head and a conditional flow-matching head.

4.4.1 Conditional flow matching head

We instantiate equation 8 using conditional flow matching. We treat the target residual as $x_1 \equiv \Delta m_t$, draw $x_0 \sim \mathcal{N}(0, I)$ of matching shape, and sample $\tau \sim \mathcal{U}(0, 1)$. Using the straight path,

$$x_\tau = (1 - \tau)x_0 + \tau \Delta m_t, \tag{12}$$

we predict $v_\theta(x_\tau, \tau \mid c_t)$, where c_t is encoded by the transformer into motion-aligned features h_t. Concretely, (x_τ, τ, h_t) are passed to an AdaLN-modulated MLP head (Li et al., 2024). The loss is

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\Big[\big\|v_\theta(x_\tau, \tau \mid c_t) - (\Delta m_t - x_0)\big\|_2^2\Big]. \tag{13}$$

At inference, we sample $\widehat{\Delta m}_t$ by integrating $\frac{dx}{d\tau} = v_\theta(x, \tau \mid c_t)$ forward from $x(0) \sim \mathcal{N}(0, I)$ using a fixed-step explicit integrator.

4.4.2 GAUSSIAN MIXTURE MODEL HEAD

Alternatively, we parameterize $p_\theta(\Delta m_t \mid c_t)$ as a Gaussian mixture with M components. Our GMM uses global mixture weights shared across all motion-token positions and position-wise Gaussian parameters:

$$p_\theta(\Delta m_t \mid c_t) = \sum_{k=1}^{M} \pi_k(c_t) \prod_{l=1}^{L} \prod_{n=1}^{N_m} \mathcal{N}\big(\Delta m_t^{(l,n)};\, \mu_k^{(l,n)}(c_t),\, \mathrm{diag}(\sigma_k^{(l,n)}(c_t)^2)\big). \tag{14}$$

A linear head on h_t^{g} predicts mixture weights $\pi(c_t)$, while a per-position head on h_t predicts $\mu_k^{(l,n)}$ and $\log \sigma_k^{(l,n)}$. We minimize

$$\mathcal{L}_{\mathrm{gmm}} = -\mathbb{E}\big[\log p_\theta(\Delta m_t \mid c_t)\big] \;-\; \lambda\, \mathbb{E}\big[H(\pi(c_t))\big]. \tag{15}$$

At inference, we sample a mixture index and then sample motion residuals from the corresponding Gaussians.

4.5 STATE-PREDICTION BASELINE

LaMo predicts future dynamics by sampling a motion residual, integrating it into a motion token, and advancing the latent state through a fixed transition model. To measure the advantage of modeling residual changes in motion space, we define a baseline that predicts the *next latent state directly*:

$$p_\theta(z_{t+1} \mid z_{t-K:t}). \tag{16}$$

For a controlled comparison, we use the same transformer configuration and context length K. The only changes are the context signal, motion-token history vs. latent-state history, and prediction target, Δm_t vs. z_{t+1}. Because latent states are denser and higher-dimensional than motion tokens, the baseline must condition on and generate a larger latent payload at each step. This difference is the variable under study rather than a handicap: dense feature forecasting is an established paradigm (Karypidis et al., 2025a; Zhou et al., 2025).

4.6 AUTOREGRESSIVE ROLLOUT

At time t, we form a context c_t from the motion-token history and current latent state, then sample

$$\widehat{\Delta m}_t \sim p_\theta(\cdot \mid c_t) \tag{17}$$

using either the flow or GMM head. The next motion token is reconstructed by

$$\hat{m}_t = \hat{m}_{t-1} + \widehat{\Delta m}_t, \tag{18}$$

and the latent state is advanced as

$$\hat{z}_{t+1} = f_{\mathrm{fwd}}(\hat{z}_t, \hat{m}_t). \tag{19}$$

Thus, information is carried through the evolving latent state $\hat{z}_t$, while stochasticity enters through sampled residual changes in motion. The motion-token context window is advanced by appending $\hat{m}_t$. For the state-prediction baseline, the latent-state window is advanced by appending $\hat{z}_{t+1}$.

Decoding latents to pixels. To visualize rollouts, we train a separate 6-layer transformer decoder d_ψ that reconstructs frames from latent states:

$$\mathcal{L}_{\mathrm{dec}} = \mathbb{E}\big[\|d_\psi(z_t) - x_t\|_2^2\big], \tag{20}$$

where x_t is the RGB frame. During evaluation, predicted latent rollouts $\{\hat{z}_t\}$ are decoded using the frozen decoder. Since L_2 objectives favor blurry averages (Mathieu et al., 2016), we also train a stronger conditional flow-matching (CFM) decoder, used for all manipulation experiments.

5 EXPERIMENTS

We evaluate LaMo on long-horizon autoregressive rollouts on BDD100K, BridgeData V2, and RoboCasa. We compare motion-residual prediction against a matched state-prediction baseline under identical transformer capacity and context length, and report rollout quality at short (0.2s), medium (0.6s), and long-term horizons (1.6s) on BDD100K (Karypidis et al., 2025b), and at 0.3s, 0.9s, and 1.8s on the manipulation datasets.



Figure 2: **Qualitative rollouts on BDD100K.** Motion-residual rollouts better preserve coordinated scene motion and background structure over long horizons than the state-prediction baseline.

5.1 Implementation

Data and preprocessing. Following prior work on latent dynamics, we first train a Midway Network to produce frozen latent states z_t and extract three levels of hierarchical motion tokens m_t, with each level consisting of ten tokens. Tokens have dimension 192 and are refined additively from coarse to fine; LaMo predicts a separate residual for each of the 3×10 token positions. We then freeze Midway Network and train probabilistic residual predictors for $p(\Delta m_t \mid m_{t-K:t-1}, z_t)$, where $\Delta m_t = m_t - m_{t-1}$. For BridgeData V2 and RoboCasa, we train a separate Midway Network per dataset (Appendix A).

Models and training setup. All probabilistic heads share the same contextual transformer backbone. We use a motion-token history of length $K{=}3$ with the current latent token z_t. The state-prediction baseline uses latent-state history with the same K. All probabilistic heads are trained with context-noise augmentation unless stated otherwise. Training is performed on $4\times$ H200 GPUs.

Flow head. For conditional flow matching, we use a 12-layer transformer backbone with model width $d = 1152$, 8 attention heads, and an AdaLN-modulated MLP flow head with 3 residual blocks. We train for 300 epochs with AdamW (Loshchilov & Hutter, 2019) at learning rate 1.5×10^{-4}. At inference, motion residuals are sampled by fixed-step ODE integration.

GMM head. For the GMM variant, we keep the same transformer backbone and predict a mixture of $M = 3$ Gaussians. Mixture weights are shared across motion-token positions, while Gaussian parameters are predicted per position. We train with GMM negative log-likelihood and an entropy regularizer using AdamW at learning rate 1.5×10^{-4} for 300 epochs.

Evaluation protocol. We evaluate under autoregressive rollout with context length $K{=}3$. For LaMo, the model predicts $\widehat{\Delta m}_t$, integrates it into $\hat{m}_t$, advances the latent state via $\hat{z}_{t+1} = f_{\text{fwd}}(\hat{z}_t, \hat{m}_t)$, and decodes predictions to pixels. For the baseline, the model directly predicts $\hat{z}_{t+1}$ from latent-state history. We report image-space metrics at $t{=}0.2$s, $t{=}0.6$s, and $t{=}1.6$s. To evaluate scene-level structure beyond pixel similarity, we also compute semantic segmentation mean Intersection-over-Union (mIoU) at the same horizons using decoded predicted frames. Segmentation details are given in Appendix B.

5.2 Results

Motion-residual prediction improves long-horizon rollouts. Our main comparison is against an architecture-matched state-prediction baseline that predicts $p_\theta(z_{t+1} \mid z_{t-K:t})$ and rolls out by feeding predicted latent states back into the context. In contrast, LaMo predicts a residual in motion space, $\widehat{\Delta m}_t \sim p_\theta(\Delta m_t \mid \hat{m}_{t-K:t-1}, \hat{z}_t)$, reconstructs $\hat{m}_t = \hat{m}_{t-1} + \widehat{\Delta m}_t$, and advances the latent

Table 1: **Long-horizon rollout quality on BDD100K.** Lower FID is better; higher PSNR/SSIM are better.

	Short-term ($t = 0.2$s)			Med-term ($t = 0.6$s)			Long-term ($t = 1.6$s)		
Method	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑
Flow (state-prediction baseline)	58	23.3	0.7295	108	20.1	0.6385	227	14.9	0.4345
GMM (motion residuals)	70	23.2	0.7277	113	**20.4**	**0.6474**	204	**16.9**	**0.5289**
Flow (motion tokens, target m, 5 steps)	61	23.3	0.7288	98	19.9	0.6165	238	15.2	0.4440
Flow (motion tokens, target m, 20 steps)	**52**	**23.7**	0.7426	74	20.2	0.6401	189	15.5	0.4665
Flow (motion tokens, target m, 50 steps)	**52**	**23.7**	**0.7427**	72	20.2	0.6395	188	15.5	0.4653
Flow (motion residuals, target Δm)	53	23.4	0.7347	**67**	20.1	0.6451	**167**	15.9	0.4953
Flow (motion residuals, 50 steps no noise)	59	23.4	0.7347	87	19.7	0.6320	221	15.4	0.4543

state with $\hat{z}_{t+1} = f_{\text{fwd}}(\hat{z}_t, \hat{m}_t)$. Table 1 shows that this residual formulation substantially improves long-horizon stability: at t=1.6s, the target-Δm flow model reduces FID from 227 for the state baseline to 167, while improving PSNR/SSIM from 14.9/0.4345 to 15.9/0.4953.

Ablating the prediction target. We next isolate the effect of the target parameterization. With the same flow-based predictor and 50 sampling steps, predicting the residual target Δm_t improves long-horizon FID from 188 to 167 compared to predicting the motion token target directly. It also improves long-horizon PSNR/SSIM from 15.5/0.4653 to 15.9/0.4953. This suggests that modeling changes in latent motion is easier to roll out over long horizons than modeling absolute motion tokens directly.

Ablating context-noise augmentation. Context-noise augmentation is important for autoregressive stability. Removing noise degrades the target-Δm flow model from 167 to 221 FID at t=1.6s, with corresponding drops in PSNR/SSIM from 15.9/0.4953 to 15.4/0.4543. This indicates that training on perturbed contexts helps the model tolerate the imperfect histories it receives during rollout.

Ablating flow sampling steps. Increasing the number of ODE integration steps improves sample quality up to a point. Moving from 5 to 20 steps substantially reduces FID across horizons, e.g., from 98 to 74 at t=0.6s and from 238 to 189 at t=1.6s. Increasing further to 50 steps gives only modest additional gains, suggesting that most of the benefit comes from using a moderately accurate flow sampler rather than very fine integration.

Flow vs. GMM. The probabilistic head controls the tradeoff between perceptual fidelity and distortion metrics. The flow head achieves the best FID, especially at medium and long horizons, while the GMM head achieves the best long-horizon PSNR/SSIM, reaching 16.9/0.5289 at t=1.6s. This is consistent with a perception–distortion tradeoff: flow sampling produces more distributionally realistic rollouts, while the GMM head better preserves pixel-level similarity under PSNR/SSIM (Blau & Michaeli, 2018).

Computational efficiency. With batch size 1 and 50 ODE steps, excluding the shared RGB decoder, LaMo requires 199 GFLOPs per predicted frame versus 749 for the state-prediction baseline, 3.76× fewer (73.4% less) while producing better rollouts.

Generalization to robot manipulation. Repeating the comparison on BridgeData V2 and RoboCasa (Table 2) under a matched backbone, decoder, and training setup, residual prediction improves FID, PSNR, and SSIM on both datasets at every horizon; at 1.8s, FID drops from 199 to 142 and from 171 to 123, roughly 28% in both cases. Appendix C further shows that the CFM decoder lowers long-horizon BDD100K FID from 167 to 137, and that doubling the history to K=6 changes results only marginally.

Qualitative rollouts preserve coherent scene evolution. Figure 2 illustrates the same trend visually. Motion-residual rollouts better preserve coherent multi-object motion and background structure, such as cars, lane markings, and signage, with fewer long-range artifacts than the state-prediction baseline.

Table 2: **Long-horizon rollout quality on robot manipulation.** Both methods share the transformer backbone, CFM RGB decoder, and training setup; they differ in context signal and prediction target.

		Short-term ($t = 0.3$s)			Med-term ($t = 0.9$s)			Long-term ($t = 1.8$s)		
Dataset	Method	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑
BridgeData V2	State-prediction baseline	49	23.8	0.7891	128	20.1	0.6423	199	16.5	0.4617
	Flow (motion residuals, target Δm)	**42**	**24.1**	**0.7987**	**84**	**20.7**	**0.6652**	**142**	**17.6**	**0.5318**
RoboCasa	State-prediction baseline	40	24.7	0.8268	103	21.4	0.6928	171	18.0	0.5394
	Flow (motion residuals, target Δm)	**35**	**24.9**	**0.8351**	**71**	**21.8**	**0.7164**	**123**	**18.9**	**0.6017**

Table 3: **Semantic segmentation quality across rollout horizons.** We report mIoU at short-, medium-, and long-term horizons. Higher is better.

Method	$t = 0.2$s	$t = 0.6$s	$t = 1.6$s
Flow + noise (target Δm)	**30.4**	**23.5**	**17.3**
Flow + noise	29.5	21.6	15.4
GMM	**30.4**	19.6	11.6
State model	29.2	15.4	8.4
No noise	29.4	20.3	13.1

5.3 Semantic segmentation

To evaluate whether predicted rollouts preserve semantically meaningful scene structure, we report semantic segmentation mIoU at the same horizons in Table 3. The target-Δm residual variant performs best across all horizons, improving long-horizon mIoU from 8.4 for the state model to 17.3. It also improves over direct flow prediction with noise, which reaches 15.4 at t=1.6s, and over the no-noise variant, which reaches 13.1. These results show that the gains from residual motion prediction are not limited to image-space metrics: the predicted rollouts also preserve scene-level semantic structure more reliably.

Scope. We compare motion-residual prediction against dense latent-state prediction under a matched transformer backbone and training setup. We therefore do not claim to outperform frontier long-video generation systems that rely on specialized long-context routing, progressive denoising, or substantially larger-scale training (Xie et al., 2024; Cai et al., 2025). Instead, our results show that, within a controlled latent-dynamics setting, predicting residual changes in motion space gives more stable long-horizon rollouts than directly predicting dense latent states. Limitations include a single, per-domain latent backbone (our manipulation results are in-domain, not zero-shot), horizons of at most 1.8s, and no downstream planning evaluation.

6 Conclusion

We introduced **LaMo**, a probabilistic latent dynamics model that predicts *motion residuals*—compact latent variables representing changes between consecutive motion tokens—instead of repeatedly regenerating dense per-frame latent states. By modeling changes in latent motion rather than dense latent states, LaMo reduces compounding drift and improves long-horizon rollouts.

On BDD100K, BridgeData V2, and RoboCasa, LaMo yields more accurate rollouts than an architecture-matched latent-state baseline at $3.8\times$ fewer FLOPs per frame, and on BDD100K it also improves semantic segmentation mIoU. Looking ahead, LaMo could enable latent dynamics planning by sampling motion-residual rollouts, integrating them into motion-token trajectories, and evaluating the resulting latent trajectories in an online MPC-style loop. Future work should train the latent dynamics model and residual predictor jointly, build the latent backbone on frozen VFM features to avoid per-domain retraining, and update the latent state with new observations during rollout to handle newly appearing objects.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing and editing code for our models, training, and evaluation, and with drafting, editing, and proofreading parts of the paper text and its LaTeX formatting. We also used generative AI tools to help identify related work and prepare bibliography entries. We have reviewed all AI-assisted work: AI-assisted code was checked by the authors, every bibliography entry was verified manually against the original publication, and all AI-assisted text was revised by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/c203d8a151612acf12457e4d67635a95-Paper.pdf.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687614. URL https://doi.org/10.1145/3680528.3687614.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *CoRR*, abs/2311.15127, 2023a. URL https://doi.org/10.48550/arXiv.2311.15127.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6228–6237, 2018. doi: 10.1109/CVPR.2018.00652. URL https://openaccess.thecvf.com/content_cvpr_2018/papers/Blau_The_Perception-Distortion_Tradeoff_CVPR_2018_paper.pdf.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. OpenAI technical report, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

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

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, Maneesh Agrawala, Lu Jiang, and Gordon Wetzstein. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025. doi: 10.48550/arXiv.2508.21058. URL https://arxiv.org/abs/2508.21058.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024a. doi: 10.48550/arXiv.2407.01392. URL https://arxiv.org/abs/2407.01392.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *NeurIPS*, pp. 6572–6583, 2018. URL http://dblp.uni-trier.de/db/conf/nips/nips2018.html#ChenRBD18.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv:2412.04445*, 2024b.

Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=vUrOuc6NR3.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in Neural Information Processing Systems*, 2016.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. In *International Conference on Machine Learning (ICML)*, 2025.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild, 2026. URL https://arxiv.org/abs/2601.05230.

Google DeepMind. Veo 3 technical report. Technical report, Google DeepMind, may 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations (ICLR)*, 2019.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXIX*, pp. 393–411, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72985-0. doi: 10.1007/978-3-031-72986-7_23. URL https://doi.org/10.1007/978-3-031-72986-7_23.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution. https://worldmodels.github.io.

Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2555–2565, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Christopher Hoang and Mengye Ren. Midway network: Learning representations for recognition and motion from latent dynamics. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZenwrTwNcj.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705v2*, 2025.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

Rico Jonschkowski, Austin Stone, Jonathan Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *European Conference on Computer Vision*, 2020.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-Foresight: Looking into the future with DINO. In *Advances in Neural Information Processing Systems*, volume 38, 2025a.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Advancing semantic future prediction through multimodal visual sequence transformers. *arXiv preprint arXiv:2501.08303*, 2025b. doi: 10.48550/arXiv.2501.08303. URL https://arxiv.org/abs/2501.08303.

Tommie Kerssies, Gabriele Berton, Ju He, Qihang Yu, Wufei Ma, Daan de Geus, Gijs Dubbelman, and Liang-Chieh Chen. A frame is worth one token: Efficient generative world modeling with delta tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27978–27988, 2026.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024. URL https://arxiv.org/abs/2406.11838.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. doi: 10.48550/arXiv.2209.03003. URL https://arxiv.org/abs/2209.03003.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations*, 2016.

R Chris Miall and Daniel M Wolpert. Forward models for physiological motor control. *Neural networks*, 9(8):1265–1279, 1996.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

Marco Pasini, Javier Nistal, Stefan Lattner, and George Fazekas. Continuous autoregressive models with noise augmentation avoid error accumulation. *arXiv preprint arXiv:2411.18447*, 2024. URL https://arxiv.org/abs/2411.18447. Accepted to NeurIPS 2024 – Audio Imagination Workshop.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Rajesh Rao and Dana Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2:79–87, 02 1999. doi: 10.1038/4580.

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

Mandyam Veerambudi Srinivasan, Simon Barry Laughlin, and A. Dubs. Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. B. Biological Sciences*, 216(1205):427–459, 11 1982. ISSN 0080-4649. doi: 10.1098/rspb.1982.0085. URL https://doi.org/10.1098/rspb.1982.0085.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning*, 2015.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024. doi: 10.48550/arXiv.2408.14837. URL https://arxiv.org/abs/2408.14837. ICLR 2025 (accepted).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual description. *ArXiv*, abs/2210.02399, 2022.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Daniel M Wolpert, Zoubin Ghahramani, and Michael I Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882, 1995.

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. *arXiv preprint arXiv:2410.08151*, 2024. URL https://arxiv.org/abs/2410.08151.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. arXiv preprint arXiv:2104.10157, 2021.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. *arXiv preprint arXiv:2505.18875*, 2025. doi: 10.48550/arXiv.2505.18875. URL https://arxiv.org/abs/2505.18875.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VYOe2eBQeh.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135, 2025.

A Robot-manipulation datasets

BridgeData V2. BridgeData V2 (Walke et al., 2023) is a large dataset of real-world robot manipulation demonstrations collected with a low-cost robot arm across 24 environments.

RoboCasa. RoboCasa (Nasiriany et al., 2024) is a large-scale simulation framework of everyday tasks in diverse kitchen scenes.

Preprocessing and evaluation. A separate Midway Network is trained on each dataset with the same architecture as for BDD100K, followed by the LaMo residual predictor and the state-prediction baseline with identical transformer configuration, context length $K{=}3$, and training recipe. Both methods are decoded with the same CFM RGB decoder. On these datasets, we report image-space metrics only.

B Semantic segmentation evaluation

We apply a pretrained semantic segmentation model to the decoded predicted frames and compute mIoU at each horizon.

C Decoder and context-length ablations

Table 4 evaluates the BDD100K flow model with target Δm under two RGB decoders and two context lengths.

RGB decoder. Replacing the 6-layer L_2 transformer decoder with the CFM decoder improves perceptual quality at all horizons, reducing long-horizon FID from 167 to 137. The slight decrease in PSNR and SSIM at longer horizons is consistent with the perception–distortion tradeoff (Blau & Michaeli, 2018). The decoder therefore contributes materially to the absolute FID values in Table 1; since the manipulation experiments use the CFM decoder for both methods, the residual-over-state advantage is not an artifact of the L_2 decoder.

Context length. K denotes the explicit motion-token history; the recurrently updated latent state $\hat{z}_t$ additionally carries information from the entire preceding trajectory. With K=3, the explicit history spans 0.6s and contains $3 \times 3 \times 10 = 90$ motion tokens, from which the transformer can infer first- and second-order changes in motion. Doubling the history to K=6 is marginally worse at 0.2s and gives only modest gains at 0.6s and 1.6s (FID improves by 2 and 4 points). We therefore use K=3, which captures most of the useful recent dynamics at lower cost.

Table 4: **Decoder and context-length ablations on BDD100K** (flow, target Δm). Lower FID is better; higher PSNR/SSIM are better.

		Short-term ($t = 0.2$s)			Med-term ($t = 0.6$s)			Long-term ($t = 1.6$s)		
RGB decoder	K	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑	FID↓	PSNR↑	SSIM↑
L_2 transformer	3	53	23.4	0.7347	67	**20.1**	**0.6451**	167	**15.9**	**0.4953**
CFM	3	**46**	**23.5**	**0.7370**	51	19.9	0.6386	137	15.6	0.4812
CFM	6	47	23.4	0.7358	**49**	20.0	0.6421	**133**	15.8	0.4880