LATENT-FORESIGHT: END-TO-END LEARNING PREDICTABLE REPRESENTATIONS FOR LATENT WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting the future evolution of a scene is a fundamental capability for world modeling. Recent work has shown that operating in the feature space of Vision Foundation Models (VFMs) yields semantically rich representations that support diverse future scene understanding tasks. However, existing approaches rely on two-stage pipelines, where VFM features are first compressed using fixed dimensionality reduction (e.g., PCA) or independently trained autoencoders, and a separate predictor is trained on top of the resulting frozen latent space. This decoupling between representation learning and temporal prediction, as well as approaches that apply predictors directly on raw VFM features, provides no guarantee that the latent space is structured for predictable dynamics. In this work, we propose `Latent-Foresight`, an end-to-end framework that jointly learns a latent tokenizer and a flow-based generative dynamics model, explicitly shaping the representation to support temporal predictability. To enable stable joint optimization, we introduce several key design choices that prevent latent collapse and align reconstruction with generative objectives. Extensive experiments show that our approach learns more temporally coherent latent representations and consistently outperforms two-stage baselines across multiple future scene understanding tasks and prediction horizons, while eliminating separate training stages, including during high-resolution adaptation.

1 INTRODUCTION

A central challenge in world modeling is to forecast the future evolution of a scene from its observed past. This capability is critical for autonomous systems, such as robots and vehicles, which must anticipate future events to plan safe and effective actions. A fundamental question underlying this problem is: *what representation should future prediction operate on?*

A common approach is to predict future world states at the pixel level, typically using diffusion models in a two-stage pipeline: an autoencoder (or tokenizer) first compresses RGB frames into a latent space, and a generative model is then trained to forecast future states in this space. While conceptually simple, such latent representations largely preserve low-level visual details, forcing the generative model to capture fine-grained appearance variations that are often irrelevant for downstream decision-making tasks. This results in high computational cost and large model requirements.

Recent work (Karypidis et al., 2025; Baldassarre et al., 2025; Zhou et al., 2025; Boduljak et al., 2025; Sun et al., 2026) has shifted toward forecasting in the feature space of Vision Foundation Models (VFMs), such as DINOv2, whose representations encode higher-level semantic structure. Operating in this space improves performance on downstream dense prediction tasks (e.g., semantic segmentation, depth estimation, surface normals) while reducing model size and computational cost. Early approaches were discriminative (Karypidis et al., 2025; Zhou et al., 2025; Baldassarre et al., 2025), relying on deterministic regression to predict future features, often after dimensionality reduction via fixed preprocessing such as PCA (Karypidis et al., 2025). However, deterministic predictors fail to capture the inherent uncertainty of future outcomes, leading to averaged predictions.

To address this limitation, recent work has introduced generative formulations (Walker et al., 2025; Boduljak et al., 2025; Sun et al., 2026), in particular flow-matching models that generate diverse

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.

future feature trajectories. Notably, VFMF (Boduljak et al., 2025) shows that compressing VFM features with an autoencoder is important for obtaining a well-conditioned latent space suitable for generative modeling. Despite these advances, existing approaches adopt a two-stage training paradigm: a tokenizer is first learned to compress VFM features, and a flow-based forecasting model is subsequently trained on the resulting frozen latent space.

While effective, this decoupled pipeline is suboptimal for generative forecasting. The latent space is optimized for reconstruction, independently of the flow-matching objective, with no guarantee that its geometry is conducive to modeling temporal dynamics. In particular, there is no reason for latent trajectories to be smooth, structured, or predictable under the learned generative flow. This raises a natural question: *can the tokenizer and the flow-matching model be trained jointly, so that the latent representation is shaped directly by the requirements of generative prediction?*

In this work, we answer this question affirmatively. We introduce `Latent-Foresight`, a framework that jointly trains a VFM feature tokenizer and a flow-matching model to forecast future latent states. To the best of our knowledge, this is the first approach to enable end-to-end training of both the tokenizer and a generative latent dynamics model in the context of VFM-based world modeling. Transitioning from a two-stage to an end-to-end formulation is, however, non-trivial. Naïvely optimizing both components jointly leads to degenerate solutions in which latent representations collapse to trivial constants, minimizing the generative loss while destroying useful structure.

We show that stable end-to-end training is achievable through simple but critical design choices, including stopping gradients through flow-matching targets and noisy inputs, normalizing the latent space, incorporating a reconstruction loss on denoised latents, and selecting the noise schedule. With these ingredients, the tokenizer and flow-matching model learn complementary objectives, yielding a latent space that is both reconstructive and well-suited for generative forecasting.

Empirically, our approach outperforms prior two-stage methods in forecasting accuracy while preserving reconstruction fidelity, using a single training stage from scratch. Beyond performance gains, our results show that reconstruction and generative objectives can be jointly optimized to shape latent representations for world modeling, yielding a simpler, more principled paradigm.

In summary, our contributions are threefold: **(1)** We show that flow-based latent world models can be trained fully end-to-end, jointly optimizing both the latent encoder and the latent flow-matching dynamics model. This establishes a new paradigm for end-to-end trainable latent world modeling, simplifying training. **(2)** To enable stable joint optimization, we introduce several key technical components that address the challenges of coupling latent representation learning with diffusion-based dynamics modeling, preventing collapse and ensuring well-structured latent spaces. **(3)** We demonstrate empirically that end-to-end training substantially improves performance over conventional decoupled pipelines, yielding consistent gains across multiple scene understanding tasks and prediction horizons.

2 Related Work

Video Prediction-RGB world models Anticipating future visual states from past observations has long been a central objective in computer vision. Early recurrent approaches based on convolutional LSTMs (Nabavi et al., 2018; Xu et al., 2018; Wang et al., 2018; Castrejon et al., 2019; Lee et al., 2021; Wu et al., 2021) established the core temporal modeling paradigm but struggled to produce sharp and temporally consistent predictions. To address this, subsequent work incorporated generative modeling through adversarial training and variational autoencoders (Yan et al., 2021; Vondrick et al., 2016; Babaeizadeh et al., 2018; Lee et al., 2018), improving diversity and visual quality of predicted frames. The introduction of diffusion models (Ho et al., 2022a;b; Harvey et al., 2022; Gao et al., 2022) further advanced prediction fidelity by modeling the full distribution over future frames. Transformer-based architectures extended these gains by capturing long-range temporal dependencies through autoregressive and masked modeling objectives (Yu et al., 2023; 2024; Gupta et al., 2023; Wang et al., 2024). Most recently, large-scale generative models such as Sora (Brooks et al., 2024) and COSMOS (Ali et al., 2025) have pushed the boundaries of photorealistic video synthesis, though at substantial computational cost. While these methods excel at generating visually realistic futures, they optimize for pixel-level appearance rather than semantic scene understanding, making them poorly suited for the needs of autonomous systems.

Latent Diffusion Models Modern video prediction methods commonly adopt latent generative approaches (Gupta et al., 2023; Yu et al., 2023; 2024; Gao et al., 2022), following a two-stage pipeline in which an autoencoder is first trained to compress frames and a generative model is subsequently trained on the frozen latent space. While this reduces computational cost and facilitates optimization, the learned representation is optimized for reconstruction rather than generation. Recent works have challenged this paradigm in image generation. REPA-E (Leng et al., 2025) enables joint training of a VAE and a diffusion transformer through representation alignment with a frozen vision encoder, accelerating convergence. UNITE (Duggal et al., 2026) jointly optimizes tokenization and generation through a shared generative encoder, while EOSTok (Chu et al., 2026) jointly trains an 1D semantic tokenizer and an autoregressive model. However, these approaches focus on static image generation. Jointly learning latent representations and generative temporal dynamics for world modeling remains largely unexplored, introducing challenges such as latent collapse and unstable latent geometry. Unlike REPA-E, which relies on an additional representation-alignment objective, our work investigates how to achieve stable end-to-end training of a tokenizer and a temporal flow-matching predictor for latent world modeling.

Semantic Future Prediction - Latent World Models DINO-Foresight (Karypidis et al., 2025) introduced forecasting in DINOv2 feature space for dense scene understanding, using PCA compression and a masked transformer predictor. DINO-WM (Zhou et al., 2025) leveraged frozen DINOv2 features for action-conditioned planning, while DINO-world (Baldassarre et al., 2025) further demonstrated the effectiveness and computational efficiency of DINOv2-based video world models. V-JEPA (Bardes et al., 2024) jointly trains an encoder and predictor through masked spatiotemporal feature prediction, using context from the entire video clip rather than only past observations, with the goal of learning visual representations rather than forecasting future states. In V-JEPA 2 (Assran et al., 2025), temporal forecasting is instead performed by training a separate predictor on frozen pretrained representations, without jointly optimizing the encoder. Frozen Forecasting (Walker et al., 2025) systematically evaluated frozen VFMs for forecasting, highlighting the advantages of video-pretrained representations. More recently, VFMF (Boduljak et al., 2025) introduced a separately trained autoencoder and generative flow matching in VFM feature space, while FlowWM (Porcher et al., 2026) performs flow matching directly on frozen DinoV3 features with a one-step projection for stable high-dimensional training. DeltaTok (Kerssies et al., 2026) proposed a compact tokenizer encoding frame-to-frame feature differences into a single token for efficient generative world modeling. LeWorldModel (Maes et al., 2026) also jointly trains an encoder and predictor from scratch, but uses discriminative (MSE-based) prediction of compact global CLS-token representations for planning on synthetic control benchmarks, rather than flow-matching-based generative prediction of dense spatial features for scene understanding. Unlike these approaches, our work jointly optimizes a VFM feature tokenizer and a generative temporal predictor end-to-end, explicitly shaping the latent space for temporal predictability.

3 METHODOLOGY

Our goal is to improve Vision Foundation Model (VFM) feature forecasting (Karypidis et al., 2025; Zhou et al., 2025; Baldassarre et al., 2025; Boduljak et al., 2025) by learning a latent representation that is explicitly shaped for generative temporal prediction. In contrast to prior approaches that rely on frozen representations (Baldassarre et al., 2025) or separately trained tokenizers (Boduljak et al., 2025; Karypidis et al., 2025), we propose a unified framework in which the latent space is learned jointly with a flow-based generative model. This allows the representation itself to adapt to the requirements of temporal forecasting rather than being fixed a priori.

3.1 MODEL COMPONENTS

VFM Feature Extraction. Following prior work (Karypidis et al., 2025), we extract dense features from multiple intermediate layers of a frozen DINOv2 ViT encoder (Oquab et al., 2024), where early layers capture local structure while deeper layers encode higher-level semantics, leading to improved performance in dense prediction tasks. Given L selected layers, we extract per-layer feature maps and concatenate them along the channel dimension to obtain a unified representation $\mathbf{F} \in \mathbb{R}^{H \times W \times D_f}$, with $D_f = L \cdot D_{\text{enc}}$, where $H \times W$ denotes the spatial resolution of the feature

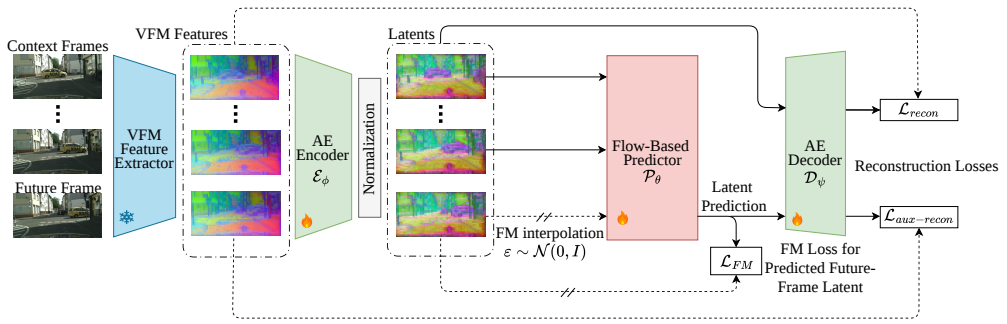


Figure 1: **Overview of `Latent-Foresight`.** We jointly train end-to-end a VFM tokenizer encoder $\mathcal{E}_\phi$, decoder $\mathcal{D}_\psi$, and flow-based predictor $\mathcal{P}_\theta$, while keeping the VFM backbone frozen. The encoder compresses per-frame features into a compact predictable latent space, followed by a normalization layer. The decoder is supervised by $\mathcal{L}_{\text{recon}}$. The predictor takes context latents and a noisy interpolation of the future latent as input, trained with $\mathcal{L}_{\text{FM}}$. Stop-gradient operations on both the future latent and the noisy input prevent collapse. $\mathcal{L}_{\text{aux-recon}}$ supervises the predicted features against the ground truth.

grid and D_{enc} is the per-layer embedding dimension. The objective of VFM forecasting is to predict the evolution of these feature maps over time.

VFM Feature Tokenizer. The resulting feature maps are high-dimensional, often exceeding several thousand channels per spatial location, which makes direct generative modeling challenging. To address this, prior work has introduced dimensionality reduction schemes such as PCA or learned autoencoders. We follow the autoencoder-based formulation and introduce an encoder $\mathcal{E}_\phi$ and decoder $\mathcal{D}_\psi$, parameterized by ϕ and ψ (implemented as light-weight vision transformers, with two transformer blocks each). The encoder maps each frame-level feature map into a compact latent representation $\mathbf{z} = \mathcal{E}_\phi(\mathbf{F}) \in \mathbb{R}^{H \times W \times C}$, with $C \ll D_f$, while the decoder reconstructs the original VFM features as $\hat{\mathbf{F}} = \mathcal{D}_\psi(\mathbf{z}) \in \mathbb{R}^{H \times W \times D_f}$. The autoencoder is trained with a per-frame reconstruction objective combining Euclidean and cosine similarity terms, following (Pan et al., 2026):

$$\mathcal{L}_{\text{recon}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{D}_\psi(\mathcal{E}_\phi(\mathbf{F}_i)) - \mathbf{F}_i\|_2^2 + (1 - \cos(\mathcal{D}_\psi(\mathcal{E}_\phi(\mathbf{F}_i)) \,,\, \mathbf{F}_i)) \,. \tag{1}$$

Flow-Based Latent Forecasting. We model temporal dynamics in the latent space using a flow-matching formulation. Given a video sequence of $N = N_c + N_p$ frames, we focus on next-frame prediction ($N_p = 1$), where the goal is to forecast the future latent $\mathbf{z}_f = \mathbf{z}_{N_c+1}$ conditioned on the context latents $\mathbf{z}_c = \{\mathbf{z}_1, \ldots, \mathbf{z}_{N_c}\}$. Longer prediction horizons are obtained autoregressively.

We define a continuous interpolation between the target latent and Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$ as

$$\mathbf{z}_f^t = t\, \mathbf{z}_f + (1 - t)\, \varepsilon, \quad t \in [0, 1], \tag{2}$$

where $t = 1$ corresponds to clean data and $t = 0$ corresponds to pure noise.

We adopt the $\mathbf{x}$-prediction parameterization of JiT (Li & He, 2025), where the predictor $\mathcal{P}_\theta$ directly estimates the clean latent: $\hat{\mathbf{z}}_f = \mathcal{P}_\theta(\mathbf{z}_f^t, t, \mathbf{z}_c)$. This parameterization has demonstrated improved stability and performance in high-dimensional generative modeling. The corresponding flow-matching objective is formulated in velocity form:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t \sim p(t),\, \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\|\hat{\mathbf{v}} - \mathbf{v}\|^2 \right], \tag{3}$$

where $\mathbf{v} = \mathbf{z}_f - \varepsilon = (\mathbf{z}_f - \mathbf{z}_f^t)/(1 - t)$ and $\hat{\mathbf{v}} = (\hat{\mathbf{z}}_f - \mathbf{z}_f^t)/(1 - t)$ denote the target and predicted velocities, respectively. The sampling distribution $p(t)$ follows a logit-normal distribution with $\text{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$, controlling the concentration of training noise levels.

3.2 End-to-End Learning

Existing latent world models operate on frozen representations, either directly from a VFM encoder or from a separately trained tokenizer. In contrast, our framework jointly learns the tokenizer and

the flow-based predictor, allowing the latent space to be shaped by the requirements of generative forecasting. This coupling ensures that the encoder does not merely compress features, but produces representations that support predictable temporal evolution under the flow model.

However, joint optimization introduces a non-trivial training dynamic. Without appropriate constraints, the model may converge to degenerate solutions where latent representations collapse to trivial constants that minimize prediction error. The goal is therefore to prevent collapse while encouraging latents that are both reconstructive and temporally structured.

Latent Normalization. Since the flow model interpolates between latent features and Gaussian noise, their scales must be compatible. We normalize the encoder $\mathcal{E}_\phi$ outputs using Batch Normalization (Ioffe & Szegedy, 2015) without affine parameters, enforcing zero-mean unit-variance latents. This prevents either signal or noise dominance during interpolation. We additionally experimented with Layer Normalization (Ba et al., 2016) and regularization objectives based on KL divergence (Kingma & Welling, 2013) and SIGReg (Balestriero & LeCun, 2025).

Stop-Gradient Strategy. To avoid trivial solutions while maintaining meaningful gradient flow, we apply stop-gradient operations in the flow objective. The velocity target is computed as

$$\mathbf{v} = \mathrm{sg}(\mathbf{z}_f) - \boldsymbol{\varepsilon}, \tag{4}$$

preventing gradients from affecting the target branch. Additionally, the noisy input is detached,

$$\hat{\mathbf{z}}_f = \mathcal{P}_\theta(\mathrm{sg}(\mathbf{z}_f^t), t, \mathbf{z}_c), \tag{5}$$

ensuring that gradients flow only through the context latents. As a result, the prediction objective shapes the encoder primarily through the context representations, encouraging latent features that support predictable temporal dynamics while preventing shortcut solutions in the noisy input latents.

Prediction Reconstruction Regularization. To preserve information in the latent space, we maintain the reconstruction objective from the tokenizer. In addition, we introduce an auxiliary reconstruction loss applied to predicted latents:

$$\mathcal{L}_{\text{aux-recon}} = \|\mathcal{D}_\psi(\hat{\mathbf{z}}_f) - \mathbf{F}_{N_c+1}\|_2^2 + 1 - \cos(\mathcal{D}_\psi(\hat{\mathbf{z}}_f), \mathbf{F}_{N_c+1}). \tag{6}$$

This term helps reduce the train-test mismatch between encoded and generated latents, while remaining secondary to the main objectives.

Full Objective. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{aux-recon}} + \mathcal{L}_{\text{FM}}, \tag{7}$$

where all components are optimized jointly over the encoder $\mathcal{E}_\phi$, decoder $\mathcal{D}_\psi$, and predictor $\mathcal{P}_\theta$.

4 Experiments

4.1 Experimental Setup

Data. We evaluate our approach on two urban driving datasets, Cityscapes (Cordts et al., 2016) and nuScenes (Caesar et al., 2020), as well as Kubric (Greff et al., 2022), a synthetic multi-object dynamics benchmark, following the evaluation protocols of (Karypidis et al., 2025; Boduljak et al., 2025). To assess scalability, we additionally train a variant, `Latent-Foresight`$^+$, on the combined Cityscapes, nuScenes, and CoVLA (Arai et al., 2025) data. More dataset details in Appendix A.2.

Implementation Details. By default, we use DINOv2-Reg with ViT-B/14 as the visual encoder, with features extracted from $L = 4$ intermediate layers and concatenated to form a $D_f = 3072$-dimensional representation per token. Our learned autoencoder $\mathcal{E}_\phi, \mathcal{D}_\psi$ incorporates spatial Rotary Position Embeddings (RoPE) and projects each per-frame feature map $\mathbf{F}_i \in \mathbb{R}^{H\times W\times D_f}$ to a compact latent $\mathbf{z}_i \in \mathbb{R}^{H\times W\times C}$ with bottleneck dimension $C = 256$. The predictor builds upon the DINO-Foresight architecture (Karypidis et al., 2025), extended with RMSNorm (Zhang & Sennrich, 2019), QK-normalization (Henry et al., 2020), and SwiGLU (Shazeer, 2020) activations for improved training stability at scale, along with spatiotemporal RoPE, and is conditioned on the flow-matching noise level t via adaLN-zero (Peebles & Xie, 2023). For evaluation, we train frozen DPT heads (Ranftl et al., 2021) for semantic segmentation, depth prediction, and surface normal estimation. In addition to our default setting, we report **Latent-Foresight+**, a scaled variant trained on

Table 1: **Comparison with state-of-the-art on high-resolution VFM forecasting** (448×896)**.** Methods that do not support a task are marked with '-'. ALL: mIoU over all classes. MO: mIoU over movable-object classes. We compare `Latent-Foresight` with `DINO-Foresight` (Karypidis et al., 2025) and `DeltaTok` (Kerssies et al., 2026) (for which results are taken from the original paper). We also include the pixel-level future generation method `VISTA` (Gao et al., 2024). $\texttt{VISTA}_{ft}$ denotes the `VISTA` model fine-tuned on Cityscapes. $\texttt{Latent-Foresight}^{+}$ denotes our `Latent-Foresight` model trained on additional data for a longer training schedule.

Method	Semantic Segmentation				Depth		Surface Normals	
	Short		Mid		Short	Mid	Short	Mid
	ALL	MO	ALL	MO	δ_1	δ_1	11.25°	11.25°
`Oracle`	77.1	77.3	77.1	77.3	89.6	89.6	96.3	96.3
$\texttt{VISTA}_{ft}$	64.9	62.1	53.9	51.0	86.4	82.8	93.0	90.0
`Dino-Foresight`	71.8	71.7	59.8	57.6	88.6	85.4	94.4	91.3
`DeltaTok`	72.1	-	60.0	-	88.5	85.6	-	-
`Latent-Foresight`	72.8	72.6	61.7	59.9	88.9	86.6	95.0	92.0
$\texttt{Latent-Foresight}^{+}$	**73.3**	**72.9**	**63.3**	**61.9**	**88.9**	**86.9**	**95.2**	**92.4**

additional data with a longer training schedule, to demonstrate the scalability of our approach. Full architecture, optimization, and training details are provided in Appendix Subsection A.3.

Evaluation Metrics. We evaluate future prediction quality across multiple scene understanding tasks. For **semantic segmentation**, we report mean Intersection over Union (mIoU) computed over all classes (ALL) and restricted to movable object classes (MO). For **depth prediction**, we report mean Absolute Relative Error (AbsRel↓) and threshold accuracy (δ_1 ↑). For **surface normals**, we report mean angular error (m↓) and the percentage of pixels with angular error below $11.25°$ ($11.25°$ ↑). In ablation studies, we additionally report the cosine similarity between reconstructed or predicted features and the ground-truth VFM features. On Cityscapes, we evaluate **short-term** and **mid-term** prediction; on nuScenes, which captures more static scenes with slower dynamics, we additionally evaluate a **longer-term** horizon. On Kubric, we follow a separate step-based protocol for comparison with prior work. Full metric definitions, movable-object classes, and per-dataset horizon lengths are provided in Appendix Subsection A.4.

Baselines. The **`Oracle`** baseline directly accesses the ground-truth future frame, providing an upper bound on performance. We additionally compare against **`VISTA`** (Gao et al., 2024), a state-of-the-art latent video diffusion world model comprising 2.5 billion parameters and trained on 1,740 hours of driving video. VISTA generates future RGB frames conditioned on past observations without action inputs. The generated frames are subsequently processed by the DINOv2-Reg encoder and frozen DPT heads for evaluation. Finally, we train and evaluate two-stage baselines in which the autoencoder-based tokenizer is first optimized independently and subsequently frozen while training the flow-based predictor. This setting follows a pipeline conceptually similar to VFMF (Boduljak et al., 2025) and enables a direct comparison with our end-to-end formulation.

4.2 Comparative Results

Comparison with State-of-the-Art on Cityscapes. Table 1 compares `Latent-Foresight` against state-of-the-art VFM forecasting methods (all metrics are reported in Appendix Table 8). Our generative framework with jointly learned latents consistently outperforms the discriminative `DINO-Foresight` (Karypidis et al., 2025) across all metrics and prediction horizons. The gains increase at longer horizons, highlighting the benefits of our approach for long-term autoregressive forecasting. Our method also surpasses the recent `DeltaTok` (Kerssies et al., 2026). The pixel-level future generation method `VISTA` (Gao et al., 2024), despite its significantly larger scale, underperforms across all tasks and horizons, highlighting the effectiveness of semantic feature forecasting for future scene understanding. Finally, scaling our approach ($\texttt{Latent-Foresight}^{+}$) with more data and longer training consistently improves all metrics, demonstrating its scalability. We further compare against the generative `VFMF` (Boduljak et al., 2025), a two-stage flow-matching approach, at 224×448 resolution in Table 2 (its only available checkpoint). We also include an ablated two-stage variant of our method (Two-Stage with AE), which isolates the effect of end-to-end training. Our method consistently outperforms `VFMF`, with additional comparisons on Kubric in Table 4.

Impact of End-to-End Latent Learning. Table 2 evaluates the impact of end-to-end latent learning on reconstruction fidelity and forecasting performance, comparing our approach against raw

Table 2: **Impact of end-to-end latent learning.** Comparison between our end-to-end formulation, two-stage pipelines based on PCA or learned autoencoders, and forecasting directly on raw VFM features. We also include the two-stage flow-matching method VFMF (Boduljak et al., 2025). We report reconstruction fidelity and forecasting performance at short- and mid-term horizons. **Cos Sim** denotes cosine similarity between predicted/reconstructed features and ground-truth VFM features. Results are reported at low and high resolution. For VFMF†, we evaluate the released checkpoint by sampling 32 flow-matching trajectories with 10 ODE steps each and averaging predicted latents. End-to-end latent learning consistently improves results while maintaining stable optimization.

Method	Dim	Reconstruction			Prediction (Short-Term)			Prediction (Mid-Term)		
		Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑
Low Resolution 224×448										
Raw VFM Features	3072	1.0	68.12	66.81	0.968	64.02	62.78	0.933	54.22	50.77
Two Stage with PCA	1152	0.977	65.63	61.97	0.955	61.56	57.93	0.921	52.49	47.60
Two Stage with PCA	256	0.952	51.19	37.18	0.940	47.85	33.63	0.914	43.10	29.46
Two-Stage with AE	256	**0.989**	67.96	66.88	0.967	64.69	62.57	0.938	56.31	53.03
VFMF†	16	0.975	66.11	64.96	0.959	62.01	60.00	0.931	51.54	45.63
End-to-End (Ours)	256	**0.989**	**68.29**	**67.26**	**0.972**	**65.45**	**64.10**	**0.944**	**56.84**	**53.86**
High Resolution 448×896										
Two Stage with AE	256	**0.991**	**77.05**	77.30	0.967	71.43	70.36	0.930	60.18	58.05
End-to-End (Ours)	256	**0.991**	77.04	**77.40**	**0.972**	**72.77**	**72.57**	**0.937**	**61.74**	**59.92**

Table 3: **Comparison across tasks at nuScenes with resolution** 448×896. We show cosine similarity between predicted and ground-truth features, and depth estimation performance (δ_1 accuracy, AbsRel error), across mid-term (9 frames, 0.75s), long-term (18 frames, 1.5s), and longer-term (27 frames, 2.25s) horizons. The last two rows evaluate zero-shot generalization: models trained on Cityscapes and directly evaluated on nuScenes.

Method	Cos Sim↑			δ_1↑			AbsRel↓		
	Mid	Long	Longer	Mid	Long	Longer	Mid	Long	Longer
Oracle	-	-	-	84.0	84.0	84.0	.138	.138	.138
Dino-Foresight	0.924	0.894	0.868	76.8	73.1	69.5	0.321	0.377	0.368
Two-Stage with AE	0.927	0.888	0.857	75.9	72.0	68.2	0.242	0.328	0.390
Latent-Foresight	**0.936**	**0.903**	**0.874**	**80.8**	**76.4**	**72.0**	**0.206**	**0.267**	**0.310**
Dino-Foresight (zero-shot)	0.905	0.865	0.837	74.8	69.5	66.0	0.349	0.424	0.415
Latent-Foresight (zero-shot)	0.920	0.876	0.848	77.2	71.8	68.2	0.306	0.387	0.392

VFM features and two-stage pipelines based on PCA or learned autoencoders (AE). Among these baselines, the learned autoencoder (Two-Stage with AE) achieves the strongest overall performance, outperforming both PCA-based compression and direct prediction on raw VFM features. In contrast, PCA at the same latent dimensionality ($C = 256$) fails to preserve sufficient feature information, substantially degrading both reconstruction and forecasting performance. These results highlight the importance of learned compression for generative forecasting. However, independently training the autoencoder leaves the latent space fixed during predictor training, limiting its adaptation to the forecasting objective. Our end-to-end formulation addresses this limitation by jointly optimizing the tokenizer and flow-based predictor, consistently improving forecasting performance while preserving reconstruction fidelity and maintaining stable training. The gains are particularly pronounced for movable-object classes (MO), suggesting that jointly learned representations better capture dynamic scene content. Moreover, the improvements over two-stage training become more pronounced after high-resolution fine-tuning (448×896), as shown on Cityscapes in Table 2 and nuScenes in Table 3. Beyond these performance gains, our single-stage end-to-end approach simplifies high-resolution adaptation, avoiding the separate fine-tuning of the autoencoder and subsequent fine-tuning of the predictor required by two-stage pipelines.

Generalization Across Driving and Non-Driving Datasets and Longer Prediction Horizons. To verify that the benefits of end-to-end latent learning extend beyond Cityscapes, we evaluate on nuScenes, a second real-world driving dataset, and Kubric, a synthetic non-driving benchmark of multi-object dynamics. Table 3 reports nuScenes results across mid-term (9 frames, 0.75s), long-term (18 frames, 1.5s), and longer-term (27 frames, 2.25s) horizons, including zero-shot transfer from models trained only on Cityscapes. Latent-Foresight consistently outperforms both DINO-Foresight and our Two-Stage with AE baseline across all metrics and horizons, with gains persisting even at 27 frames, demonstrating its effectiveness for long autoregressive rollouts. In the zero-shot setting, Latent-Foresight outperforms DINO-Foresight across all metrics, suggest-

ing that the learned latent representations transfer across driving datasets. Table 4 evaluates generalization beyond driving on Kubric. With a single generation, `Latent-Foresight` achieves higher foreground IoU than VFMF at both 1-step and 8-step horizons, even compared to its 32-generation setting, while requiring substantially less inference compute. These results demonstrate that the benefits of end-to-end latent learning extend beyond autonomous driving.

4.3 Experimental analysis

Training Recipe Ablation. Table 5 evaluates the key components of our end-to-end training strategy. Our full model (a) achieves the strongest overall performance across reconstruction and forecasting. Removing RoPE from the autoencoder (b) has only a minor impact, whereas removing the auxiliary reconstruction loss $\mathcal{L}_{\text{aux-recon}}$ (c) consistently degrades forecasting performance, particularly at longer horizons. This supports the role of auxiliary reconstruction in reducing the train–test mismatch between encoded and predicted latents. *Latent normalization is essential for stable training.* Replacing Batch Normalization with Layer Normalization (d) preserves competitive reconstruction quality but degrades forecasting performance, particularly at longer horizons. KL regularization toward a unit Gaussian (e) substantially degrades both reconstruction and forecasting, whereas SIGReg (f) performs comparably to Batch Normalization. We therefore retain Batch Normalization for its simplicity and effectiveness. Removing latent normalization entirely (g) leads to training collapse, highlighting the importance of controlling the latent distribution during end-to-end flow matching. *Finally, the stop-gradient operations play distinct roles.* Removing the stop-gradient on the target latent (h) causes training collapse, as it prevents the prediction objective from destabilizing the encoder. Removing the stop-gradient on the noisy latent (i) preserves training stability but degrades forecasting performance, suggesting that gradients propagated through the noisy interpolation interfere with learning predictive representations.

Table 4: **Comparison with VFMF on Kubric** (224×224), following the VFMF evaluation protocol (4 conditioning frames, timestep of 2). VFMF results are averaged over 32 generated samples following the original paper; `Latent-Foresight` requires only a single generation. VFMF* denotes results as originally reported in the paper.

Method	1-step			8-step		
	BG↑	ALL↑	MO↑	BG↑	ALL↑	MO↑
VFMF (1 gen)	97.71	88.19	78.66	90.34	59.53	28.71
VFMF*(32 gen)	–	–	–	91.62	61.74	31.86
VFMF (32 gen)	98.09	89.99	81.88	**91.95**	62.07	32.19
`Latent-Foresight` (1 gen)	**98.32**	**91.16**	**83.99**	91.64	**63.47**	**35.29**

Bottleneck Dimensionality. Table 6 examines the effect of the latent bottleneck dimension C. While reconstruction cosine similarity improves monotonically with dimensionality, forecasting performance is strongest at intermediate dimensions ($C \in [128, 512]$). $C = 1152$ achieves the highest reconstruction similarity (only marginally above $C \in [256, 512]$) but substantially degrades mid-term forecasting, highlighting the trade-off between reconstruction fidelity and temporal predictability. Conversely, excessively small bottlenecks ($C = 32$) impair both reconstruction and forecasting. We select $C = 256$, as it provides near-optimal reconstruction and strong forecasting at both horizons (all metrics are reported in Appendix Table 10).

Noise Level Distribution. Table 7 examines the effect of the noise level distribution $p(t)$. While reconstruction quality remains largely unchanged, forecasting performance benefits from shifting the logit-normal distribution toward higher noise levels ($\mu = -2$), compared to uniform sampling

Table 5: **Ablation of the end-to-end training strategy.** We analyze the contribution of the main components enabling stable end-to-end latent learning, including the auxiliary reconstruction objective, latent normalization scheme, and stop-gradient operations used in the flow-matching objective. All results are reported at resolution 224×448. Removing latent normalization or the stop-gradient on the target latent leads to training collapse, highlighting their importance for stable optimization.

Method	Reconstruction			Prediction (Short-Term)			Prediction (Mid-Term)		
	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑
(a) `Latent-Foresight` (Ours)	**0.989**	**68.29**	**67.26**	**0.972**	**65.45**	64.10	**0.944**	**56.84**	**53.86**
(b) w/o RoPE in AE	**0.989**	68.22	**67.26**	0.971	65.42	**64.37**	**0.944**	56.60	53.42
(c) w/o Auxiliary Reconstruction ($\mathcal{L}_{\text{aux-recon}}$)	0.988	68.27	67.19	0.967	65.09	63.43	0.937	56.55	53.36
(d) BatchNorm → LayerNorm	0.989	68.27	67.22	0.970	64.70	63.53	0.942	55.88	52.66
(e) BatchNorm → KL Regularization	0.968	64.43	64.50	0.955	61.33	61.31	0.913	46.64	41.78
(f) BatchNorm → SIGReg	0.988	68.19	67.03	0.971	**65.45**	64.22	**0.944**	56.60	53.46
(g) w/o Latent Normalization	*Training Collapse*								
(h) w/o Stop-Grad on Target Latent ($\text{sg}(\mathbf{z}_f)) - \boldsymbol{\varepsilon}$	*Training Collapse*								
(i) w/o Stop-Grad on Noisy Latent ($\text{sg}(\mathbf{z}_f^t)$)	0.987	67.97	66.52	0.966	63.17	61.47	0.937	54.48	51.33

Table 6: **Ablation of Bottleneck Dimensionality.** We vary the latent dimension C of the autoencoder and report reconstruction fidelity and prediction quality at short- and mid-term horizons. All results are reported at 224×448 resolution and do not include RoPE in the AE.

C	Reconstruction Cos Sim↑	Prediction MO↑ Short-term	Mid-term
32	0.980	59.93	49.82
64	0.984	62.48	54.03
128	0.986	64.03	54.12
256	0.989	64.37	53.42
512	0.990	64.00	54.24
1152	0.991	63.98	51.99

Table 7: **Ablation of Noise Level Distribution $p(t)$.** We compare uniform sampling and logit-normal distributions with different mean and standard deviation parameters for training noise levels. All results are reported at 224×448 resolution, do not include RoPE in the autoencoder, and use a latent dimension of $C = 512$.

Noise Distribution	Recon. Cos Sim↑	Prediction MO↑ Short-term	Mid-term
Uniform	0.990	63.11	51.00
Logit-Normal (-0.8,0.8)	0.990	62.75	50.89
Logit-Normal (-2,0.8)	0.990	64.40	53.42
Logit-Normal (-2,1.5)	0.990	64.00	54.24

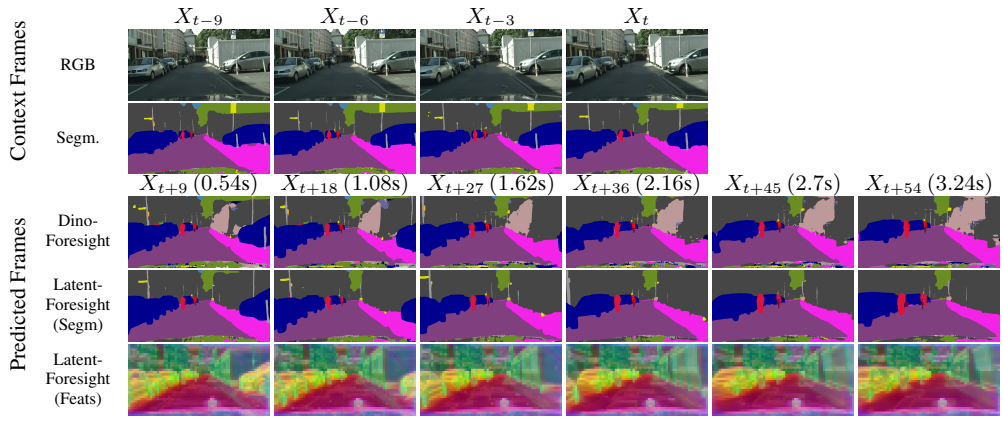


Figure 2: **Qualitative comparison of long-term forecasting.** Given four context frames (top), we forecast up to 3.24s ahead. We show semantic segmentation from `DINO-Foresight` and `Latent-Foresight`, and a PCA visualization of `Latent-Foresight` predicted features.

and the JiT default ($\mu = -0.8$, $\sigma = 0.8$). Increasing its standard deviation to $\sigma = 1.5$ further improves mid-term segmentation performance. We therefore adopt the logit-normal distribution with $(\mu = -2, \sigma = 1.5)$ as our default (all metrics are reported in Appendix Table 11).

Qualitative results. Figure 2 compares long-term forecasts of `DINO-Foresight` and `Latent-Foresight` up to 3.24s ahead. Beyond roughly 2s, `DINO-Foresight` predictions become nearly static, retaining segmentation artifacts such as spurious regions and noisy segments near the image border. In contrast, `Latent-Foresight` captures the apparent motion of cars and roadside structures under ego-motion, while better preserving scene layout and object boundaries. Small objects, including pedestrians and cyclists, also remain more clearly delineated across prediction horizons and evolve more plausibly as the ego-vehicle moves. The PCA visualization of predicted features exhibits similar temporal evolution, suggesting that these dynamics are captured in the predicted feature representations.

5 Conclusion

We presented `Latent-Foresight`, an end-to-end framework for VFM-based world modeling that jointly learns a feature tokenizer and a flow-based temporal predictor, explicitly shaping latent representations for temporal predictability. We identified simple but critical design choices that prevent latent collapse and enable stable joint optimization. Extensive experiments demonstrate consistent improvements over conventional two-stage approaches across multiple future scene understanding tasks and prediction horizons. Overall, this work establishes end-to-end latent world modeling as a simple and effective alternative to decoupled pipelines, avoiding separate tokenizer and predictor training stages even during high-resolution adaptation. It also highlights learning representations under temporal prediction objectives as a promising direction for world modeling.

AI USE STATEMENT

We used generative AI tools to polish writing, fix grammar, and help construct and align figures and tables, including generating LaTeX for qualitative comparison figures and ablation tables. We did not use AI tools for research ideation, experimental design, or writing the core scientific content. All AI-assisted output was reviewed by the authors: text edits were checked for accuracy, and figure/table code was verified and compiled to confirm correct rendering. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, new dataset releases, or other concerns raised in the ICLR Code of Ethics; all datasets used are existing public benchmarks used under their respective licenses.

REPRODUCIBILITY STATEMENT

We provide implementation details for the autoencoder and predictor architectures, optimization, and training recipe in Section 4.1 and Appendix A.3. Dataset descriptions and preprocessing are given in Appendix A.2, and evaluation metrics and prediction horizons are detailed in Appendix A.4. Hyperparameter choices are further supported by the ablation studies in Section 4.3 and Appendix A.1. Code and model checkpoints will be made publicly available upon acceptance..

REFERENCES

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *WACV*, 2025.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018. URL https://openreview.net/forum?id=rk49Mg-CW.

Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: Dino as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Gabrijel Boduljak, Yushi Lan, Christian Rupprecht, and Andrea Vedaldi. Vfmf: World modeling by forecasting vision foundation model features, 2025. URL https://arxiv.org/abs/2512.11225.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *CVPR*, 2019.

Wenda Chu, Bingliang Zhang, Jiaqi Han, Yizhuo Li, Linjie Yang, Yisong Yue, and Qiushan Guo. End-to-end autoregressive image generation with 1d semantic tokenizer, 2026. URL https://arxiv.org/abs/2605.00503.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, June 2016.

Shivam Duggal, Xingjian Bai, Zongze Wu, Richard Zhang, Eli Shechtman, Antonio Torralba, Phillip Isola, and William T Freeman. End-to-end training for unified tokenization and latent denoising. *arXiv preprint arXiv:2603.22283*, 2026.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *NeurIPS*, 2024. URL https://openreview.net/forum?id=Tw9nfNyOMy.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *CVPR*, 2022.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *CVPR*, 2022.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. In *ICLR*, 2023. URL https://openreview.net/forum?id=QAV2CcLEDh.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Dietrich Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *NeurIPS*, 2022. URL https://openreview.net/forum?id=0RTJcuvHtIu.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/39235c56aef13fb05a6adc95eb9d8d66-Paper-Conference.pdf.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-foresight: Looking into the future with DINO. In *NeurIPS*, 2025. URL https://openreview.net/forum?id=gimtybo07H.

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. *CVPR*, 2026.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *CVPR*, 2021.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *CVPR*, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

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

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Seyed Shahabeddin Nabavi, Mrigank Rochan, and Yang Wang. Future semantic segmentation with convolutional lstm. In *BMVC*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Yueming Pan, Ruoyu Feng, Qi Dai, Yuqi Wang, Wenfeng Lin, Mingyu Guo, Chong Luo, and Nanning Zheng. Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. In *CVPR*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Francois Porcher, Nicolas Carion, Karteek Alahari, and Shizhe Chen. Flow matching in feature space for stochastic world modeling. *arXiv preprint arXiv:2606.29059*, 2026.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *CVPR*, 2021.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Xiangyu Sun, Shijie Wang, Fengyi Zhang, Lin Liu, Caiyan Jia, Ziying Song, Zi Huang, and Yadan Luo. Vggt-world: Transforming vggt into an autoregressive geometry world model. *arXiv preprint arXiv:2603.12655*, 2026.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016.

Jacob C Walker, Pedro Vélez, Luisa Polania Cabrera, Guangyao Zhou, Rishabh Kabra, Carl Doersch, Maks Ovsjanikov, João Carreira, and Shiry Ginosar. Generalist forecasting with frozen video models via latent diffusion. *arXiv preprint arXiv:2507.13942*, 2025.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *ICLR*, 2018.

Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. Motionrnn: A flexible model for video prediction with spacetime-varying motions. In *CVPR*, 2021.

Jingwei Xu, Bingbing Ni, Zefan Li, Shuo Cheng, and Xiaokang Yang. Structure preserving video prediction. In *CVPR*, June 2018.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024b. URL https://openreview.net/forum?id=cFTi3gLJ1X.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *CVPR*, 2023.

Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *ICLR*, 2024. URL https://openreview.net/forum?id=gzqrANCF4g.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *ICML*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

A Appendix

A.1 Additional Results.

Full Comparison on Cityscapes. Table 8 extends Table 1 with all metrics, including depth AbsRel and surface normal mean angular error, and adds a Copy Last baseline that repeats the last context frame. The sharp drop of Copy Last at the mid-term horizon confirms that the benchmark requires modeling non-trivial scene dynamics. `Latent-Foresight` outperforms `DINO-Foresight` on semantic segmentation, depth δ_1, and both surface normal metrics at both horizons, with the largest gains at the mid-term horizon. For depth AbsRel, both methods perform comparably.

Reconstruction Loss. Table 9 compares reconstruction objectives for the autoencoder. Using only the cosine similarity term fails to preserve feature information, substantially degrading both reconstruction and forecasting, likely because the scale-invariant cosine loss leaves feature magnitudes unconstrained. MSE alone and the combination of MSE and cosine similarity perform comparably; we adopt the combined objective following Pan et al. (2026).

Bottleneck Dimensionality and Noise Level Distribution. Tables 10 and 11 report all metrics for the ablations in Tables 6 and 7. Segmentation over all classes (ALL) and predicted-feature cosine similarity follow the same trends as the movable-object mIoU discussed in Section 4: forecasting is strongest at intermediate bottleneck dimensions and degrades at $C = 1152$, while shifting the logit-normal distribution to $\mu = -2$ improves forecasting over uniform sampling and the JiT default, with $\sigma = 1.5$ yielding the best mid-term performance.

Table 8: **Comparison across tasks on Cityscapes**. We use DINOv2 encoder and show performance on segmentation (ALL, MO), depth estimation (δ_1 accuracy, AbsRel error), and surface normal prediction (m, percentage within 11.25°). VISTA$_{ft}$ is the VISTA model fine-tuned on Cityscapes.

Method	Semantic Segm.				Depth				Surface Normals			
	Short		Mid		Short		Mid		Short		Mid	
	ALL↑	MO↑	ALL↑	MO↑	δ_1 ↑	AbsR↓	δ_1↑	AbsR↓	m↓	11.25°↑	m↓	11.25°↑
Oracle	77.1	77.3	77.1	77.3	89.6	.103	89.6	.103	2.88	96.3	2.88	96.3
Copy Last	54.7	52.0	40.4	32.3	84.1	.154	77.8	.212	4.41	89.2	5.39	84.0
VISTA$_{ft}$	64.9	62.1	53.9	51.0	86.4	.124	82.8	.153	3.75	93.0	4.30	90.0
Dino-Foresight	71.8	71.7	59.8	57.6	88.6	**.114**	85.4	**.136**	3.39	94.4	4.00	91.3
Latent-Foresight	72.8	72.6	61.7	59.9	88.9	.118	86.6	.137	3.13	95	3.77	92
Latent-Foresight$^+$	**73.3**	**72.9**	**63.3**	**61.9**	**88.9**	.120	**86.9**	**.136**	**3.11**	**95.2**	**3.68**	**92.4**

Table 9: **Ablation of Reconstruction Loss.** We vary the reconstruction objective used to train the autoencoder and report reconstruction fidelity and prediction quality at short- and mid-term horizons. All results are reported at 224×448 resolution.

Method	Reconstruction			Prediction (Short-Term)			Prediction (Mid-Term)		
	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑
Cosine Only	0.889	27.50	15.78	0.895	33.22	24.23	0.881	32.46	22.92
MSE Only	0.988	68.26	67.28	0.971	65.36	64.14	0.943	56.54	53.59
MSE + Cosine	0.989	68.22	67.26	0.971	65.24	64.37	0.944	56.60	53.42

A.2 Datasets.

Cityscapes (Cordts et al., 2016) provides 2,975 training and 500 validation video sequences captured at 16 fps with a resolution of 1024×2048 pixels. Each sequence consists of 30 frames, with the 20th frame annotated for semantic segmentation across 19 classes. **nuScenes** (Caesar et al., 2020) comprises 700 training and 150 validation scenes recorded at 12 Hz, with each scene spanning 20 seconds of urban driving. **Kubric** (Greff et al., 2022) is a synthetic benchmark of multi-object scenes with controlled dynamics, which we use to evaluate generalization beyond the driving domain. We use the MOVi-A configuration, comprising 9,703 training and 250 validation sequences captured at 12 fps with a resolution of 256×256 pixels. Each sequence consists of 24 frames depicting 3–10 rigid objects moving on a static background with collisions, with full per-frame annotations (segmentation, depth, flow, 3D) available. **CoVLA** (Arai et al., 2025) is a real-world driving dataset comprising 10,000 video sequences spanning over 80 hours, captured at 20 Hz, with accompanying vehicle state, trajectory, and scene annotations. We use CoVLA alongside Cityscapes and nuScenes to train Latent-Foresight$^+$, demonstrating that our approach benefits from larger and more diverse driving data.

A.3 Implementation Details

Autoencoder and Predictor Architecture. The autoencoder $\mathcal{E}_\phi, \mathcal{D}_\psi$ consists of a transformer encoder and decoder, each with 2 layers, a hidden dimension of 1536, and 6 attention heads. The predictor operates on the latent space defined by the autoencoder and consists of 12 layers with a hidden dimension of $d = 1152$, processing sequences of $N = 5$ frames ($N_c = 4$ context frames and $N_p = 1$ future frame). The noise level t is injected via an adaLN-zero conditioning module, which produces a shared set of modulation parameters applied across all transformer layers.

Optimization and Training. For end-to-end training, we use the AdamW optimizer (Loshchilov & Hutter, 2019) with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay 0.2, and a learning rate of 6.4×10^{-4} with cosine annealing. Following Karypidis et al. (2025), for driving datasets we train at low resolution and fine-tune at high resolution: models are pretrained at 224×448 and subsequently fine-tuned at 448×896 for evaluation against prior work. For Kubric, we train only at 224×224 resolution for 800 epochs. Training is conducted on 8 GH200 GPUs with an effective batch size of 64.

Table 10: **Ablation of Bottleneck Dimensionality.** We vary the latent dimension C of the autoencoder and report reconstruction fidelity and prediction quality at short- and mid-term horizons. All results are reported at 224×448 resolution and do not include RoPE in the autoencoder.

Dimensionality	Reconstruction			Prediction (Short-Term)			Prediction (Mid-Term)		
	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑
32	0.980	64.90	61.84	0.966	62.59	59.93	0.941	54.37	49.82
64	0.984	67.14	65.79	0.969	64.24	62.48	0.943	56.57	54.03
128	0.986	67.77	66.51	0.971	65.26	64.03	0.945	56.95	54.12
256	0.989	68.22	67.26	0.971	65.42	64.37	0.944	56.60	53.42
512	0.990	68.21	67.18	0.971	65.23	64.00	0.942	56.80	54.24
1152	0.991	68.08	66.94	0.970	64.95	63.98	0.936	55.07	51.99

Table 11: **Ablation of Noise Level Distribution $p(t)$.** We compare uniform sampling and logit-normal distributions with different mean and standard deviation parameters for training noise levels. All results are reported at 224×448 resolution, do not include RoPE in the autoencoder, and use a latent dimension of $C = 512$.

Noise Distribution	Reconstruction			Prediction (Short-Term)			Prediction (Mid-Term)		
	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑	Cos Sim↑	ALL↑	MO↑
Uniform	0.990	68.17	67.17	0.969	64.34	63.11	0.937	54.60	51.00
Logit-Normal (-0.8,0.8)	0.990	68.18	67.18	0.967	63.73	62.75	0.933	53.94	50.89
Logit-Normal (-2,0.8)	0.990	68.22	67.18	0.971	65.35	64.40	0.941	56.30	53.42
Logit-Normal (-2,1.5)	0.990	68.21	67.18	0.971	65.23	64.00	0.942	56.80	54.24

Downstream heads. We provide implementation details for the heads trained on different downstream tasks. We train frozen DPT heads (Ranftl et al., 2021) for semantic segmentation, depth estimation, and surface normal prediction, using the implementation from Depth Anything (Yang et al., 2024a;b) with feature dimensionality 256 and `dpt_out_channels = [128, 256, 512, 512]`. All heads are trained for 100 epochs with a batch size of 128 on 16×8 GPUs, using AdamW with a learning rate of 1.6×10^{-3}, linear warmup for the first 10 epochs, and weight decay 10^{-4}. For semantic segmentation, we use a polynomial learning rate schedule and cross-entropy loss over 19 classes. For depth estimation, we use cosine annealing and cross-entropy loss over 256 discretized depth bins. For surface normal estimation, we use a polynomial schedule and a combined cosine similarity and L_2 loss with weighted averaging.

A.4 Definitions of Evaluation Metrics

Movable Object Classes. For semantic segmentation, movable object (MO) classes comprise person, rider, car, truck, bus, train, motorcycle, and bicycle.

Prediction Horizons. On **Cityscapes**, we report short-term (3 frames, $\sim$0.18s) and mid-term (9 frames, $\sim$0.54s) prediction. On **nuScenes**, which captures more static scenes with slower dynamics, we report mid-term (9 frames, $\sim$0.75s), long-term (18 frames, $\sim$1.5s), and longer-term (27 frames, $\sim$2.25s) prediction. On **Kubric**, following the VFMF (Boduljak et al., 2025) evaluation protocol, we report 1-step ($\sim$0.17s) and 8-step ($\sim$1.33s) prediction.

For **depth prediction**, we report mean Absolute Relative Error AbsRel $= \frac{1}{M}\sum_{i=1}^{M}\frac{|a_i - b_i|}{b_i}$, where a_i and b_i are the predicted and ground truth depths at pixel i, and threshold accuracy δ_1, the percentage of pixels satisfying $\max\left(\frac{a_i}{b_i}, \frac{b_i}{a_i}\right) < 1.25$. For **surface normal prediction**, we report the mean angular error m $= \frac{1}{N}\sum_{i=1}^{N}\cos^{-1}\left(\frac{\mathbf{n}_i \cdot \tilde{\mathbf{n}}_i}{\|\mathbf{n}_i\|\|\tilde{\mathbf{n}}_i\|}\right)$, where $\mathbf{n}_i$ and $\tilde{\mathbf{n}}_i$ are the predicted and ground truth normals, and the percentage of pixels with angular error below $11.25°$, computed as $\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\theta_i < 11.25°)$.

A.5 Additional Visualizations

We provide additional qualitative comparisons between Latent-Foresight and the two-stage baselines, Dino-Foresight and VFMF, on Cityscapes validation scenes. Figure 3 and Figure 4 show predictions for two representative scenes, at both short-term (a) and mid-term (b) horizons, across semantic segmentation, depth, and surface normal prediction. Across both scenes, Latent-Foresight consistently produces better predictions than the two-stage baselines: for segmentation and depth, it more accurately predicts the position and motion of dynamic objects, while for surface normals, it yields sharper predictions.

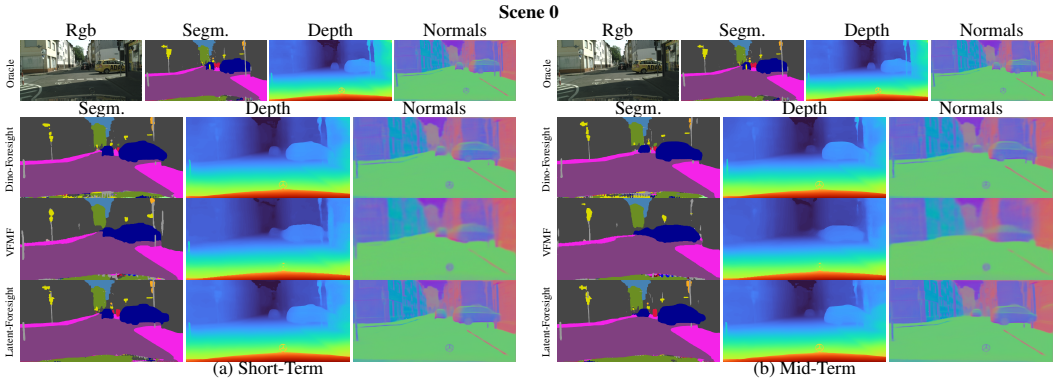


Figure 3: **Qualitative comparison on Scene 0.** We compare Dino-Foresight, VFMF, and Latent-Foresight (Ours) against ground truth (GT) for semantic segmentation, depth, and surface normal prediction at short-term (a) and mid-term (b) horizons. Latent-Foresight yields predictions that more closely match the oracle, particularly around dynamic foreground objects.

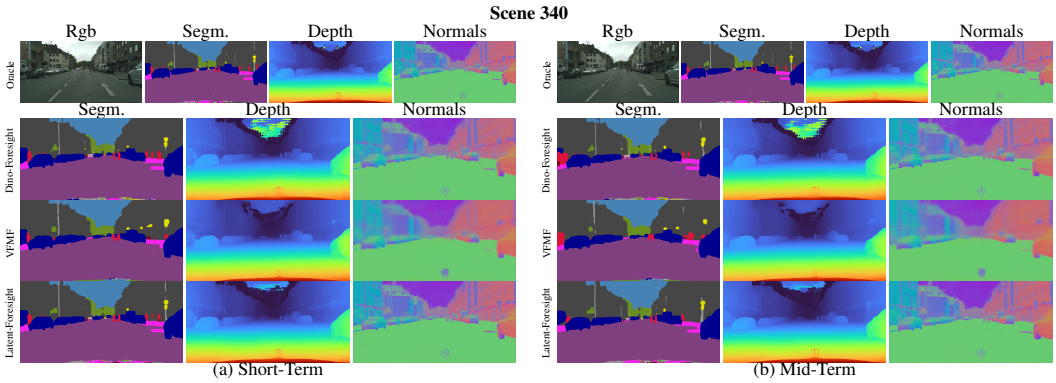


Figure 4: **Qualitative comparison on Scene 340.** We compare Dino-Foresight, VFMF, and Latent-Foresight (Ours) against ground truth (GT) for semantic segmentation, depth, and surface normal prediction at short-term (a) and mid-term (b) horizons. Latent-Foresight produces sharper and more accurate predictions than the two-stage baselines across all three tasks.

B Limitations and Future Work

While our end-to-end formulation focuses on the joint training of tokenizer and flow-based predictor, the autoencoder architecture itself is kept lightweight. Future work could explore more expressive tokenizer architectures, as well as strategies for reducing the number of spatial tokens, for instance through learned spatial pooling or cross-attention-based compression, which would reduce the sequence length seen by the predictor and enable more computationally efficient generative modeling at higher resolutions.

In addition, although the current framework focuses on next-frame prediction, the model design naturally lends itself to broader temporal modeling strategies. A natural extension would be to jointly model multiple future frames in a single forward pass, for instance through multi-step flow matching or latent sequence generation.

C Broader Impact

Our work advances efficient and scalable semantic future prediction by jointly learning a compact latent representation and a generative predictor over semantically rich VFM features. The resulting framework integrates flexibly with diverse scene understanding tasks without retraining, making it directly applicable to domains such as autonomous driving and robotics. By replacing two-stage pipelines with a single end-to-end training paradigm, our approach also reduces the complexity and overhead required to develop and deploy world models. We do not foresee direct misuse risks in our work. However, as our method builds upon pretrained VFMs, any biases embedded in those models may propagate into the predicted semantic representations, potentially affecting downstream decisions in deployment scenarios. We encourage practitioners to carefully evaluate such biases before deploying systems that rely on future predictions for real-world decision-making.