STATES BEFORE PIXELS: LARGE-SCALE TEXT-TO-VIDEO GENERATION IN REPRESENTATION SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Video representations learned by vision foundation models encode scene semantics and dynamics as spatiotemporal world states beyond pixel appearance. While representation autoencoders have shown promise for image generation, representation-space video generation remains largely underexplored, with their higher dimensionality posing additional challenges. In this paper, we demonstrate that native video representations can serve as effective modeling targets for large-scale, open-domain text-to-video generation. Our approach, ***WorldFlow***, separates world-state modeling from visual rendering: a text-conditioned flow model first generates latent world states directly in native V-JEPA space, and a second-stage flow model then renders these states into high-quality videos in VAE space. This design retains the learned semantic structure of pretrained video representations as the basis for generation, while allowing fine visual details to be synthesized in a space designed for reconstruction. To enable effective diffusion-transformer denoising in this high-dimensional representation space, we systematically study its geometry and prediction parameterization. We find that by conducting Riemannian Flow Matching on the native representation manifold with $\boldsymbol{v}$-prediction, high-dimensional video representations can be effectively modeled without introducing additional compression. Extensive experiments demonstrate that WorldFlow achieves high-quality text-to-video generation. Beyond RGB, the same generated world states can be readily rendered into well-aligned multimodal outputs such as depth, segmentation, human pose, *etc*. These results demonstrate the feasibility of generating unified world states containing appearance, semantics, geometry, and dynamics within a shared representation.

1 INTRODUCTION

Following the success of latent diffusion models (Rombach et al., 2022), variational autoencoder (VAE) latents have become the standard modeling space for high-quality image and video generation (Blattmann et al., 2023b;a; Yang et al., 2025; Team Wan et al., 2025). Optimized for reconstruction, they preserve visual detail in a compact form. More recently, representations from vision foundation models (VFMs) have offered an alternative basis for generation, both as supervision (Yu et al., 2025) and as direct targets (Zheng et al., 2025; Singh et al., 2026): in image generation, representation autoencoders (RAE) (Zheng et al., 2025) show that semantically structured representations support high-quality reconstruction and accelerate training.

Extending this paradigm to video is explored in concurrent work (Guo et al., 2026; Xie et al., 2026). Unlike reconstruction-oriented VAE latents, predictive video representations capture compact world states with rich semantics while discarding unpredictable pixel detail (LeCun, 2022; Bardes et al., 2024). Our feature similarity analysis (Fig. 5) reflects this: V-JEPA emphasizes high-level object structure while VAE latents follow color and texture. Faithful RGB reconstruction, however, needs appearance information that predictive representations may not retain, consistent with the lower PSNR reported for decoding from video representations despite competitive reconstruction FVD (Guo et al., 2026). These observations suggest combining the strengths of both spaces: first generate semantically rich world states in representation space, then synthesize their visual appearance through conditional generation in VAE space.

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.

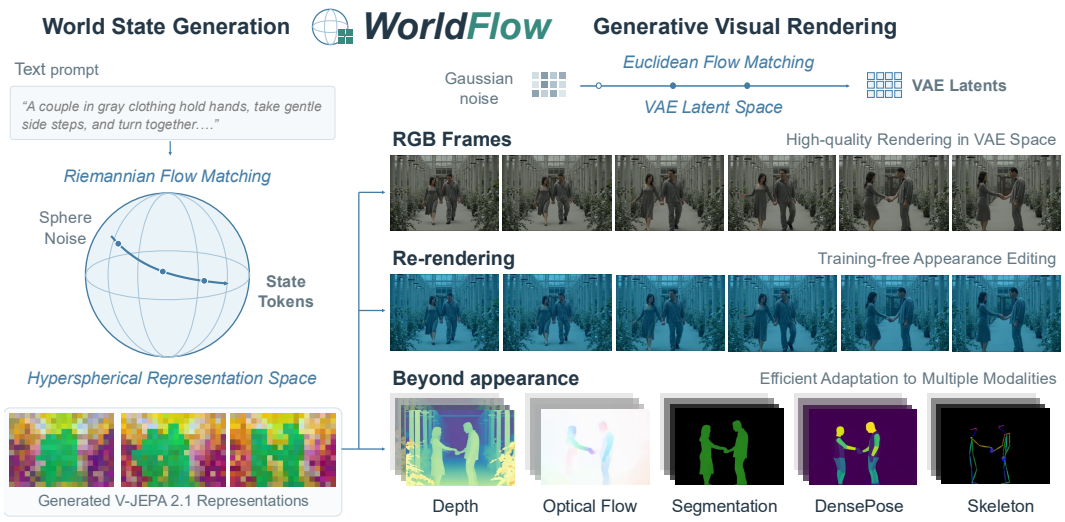

Figure 1: **WorldFlow generates video in two stages.** Stage one (WorldFlow-S1) generates latent world states in native V-JEPA representation space via Riemannian flow matching on a tokenwise hypersphere; stage two (WorldFlow-S2) renders them into RGB video via conditional flow matching in VAE space. The generated states encode structural and temporal information independent of appearance, enabling consistent synthesis across diverse prompts while supporting downstream applications such as appearance editing and multimodal rendering.

Building on this motivation, we propose ***WorldFlow***, a two-stage framework that *generates latent world states before rendering their visual appearance*. A first-stage text-conditioned flow model directly generates world states in native V-JEPA 2.1 representation space (Mur-Labadia et al., 2026), aiming to leverage its rich semantic organization to model scene structure and dynamics. A second-stage conditional flow model then operates in the reconstruction-oriented VAE space, using both the generated world states and text prompt to synthesize fine appearance details not fully determined by the states. This paradigm assigns state modeling and visual rendering to spaces suited to their respective roles, combining the merits of both worlds.

Generating world states directly poses a further challenge: native spatiotemporal representations are high-dimensional, leading recent methods to introduce compression. Our analysis shows this changes more than dimensionality — compressed V-JEPA exhibits similarity patterns closer to VAE latents, with weaker object-level associations (Fig. 5). To retain the original organization that motivates their use for state generation, we investigate how to model these representations through flow matching directly in their *native, high-dimensional* space. Specifically, we consider both flow geometry and prediction parameterization. Like most VFM representations, V-JEPA tokens lie on a hyperspherical manifold induced by normalization operations. Building on Riemannian flow matching (Chen & Lipman, 2024; Kumar & Patel, 2026) and JiT (Li & He, 2026), we systematically compare Euclidean and hyperspherical flow paths under both $\boldsymbol{v}$-prediction and $\boldsymbol{x}$-prediction. Our findings lead us to adopt hyperspherical flow matching with $\boldsymbol{v}$-prediction: at each sampling step, the model predicts a tangent velocity at the current state and uses it to guide an update along the same manifold. This formulation yields significantly faster convergence than the standard Euclidean $\boldsymbol{v}$-prediction baseline, making native video representation generation tractable.

We evaluate WorldFlow for large-scale, open-domain text-to-video generation, adapting pretrained LTX Video diffusion transformers for both stages. Our experiments demonstrate that native video representation can serve as effective generation targets for high-quality video synthesis, improving consistency and spatial quality in generated videos.

Moreover, WorldFlow models rich world states *beyond RGB appearance*: the same generated states can be rendered into aligned depth, optical flow, and human pose with lightweight adaptation, and support structure-preserving re-rendering by editing the rendering prompt alone.

We summarize our contributions as follows:

- We propose ***WorldFlow***, a two-stage framework that generates latent world states in native representation space and renders their visual appearance in VAE space, achieving high-quality text-to-video synthesis with improved consistency.
- We develop an effective flow matching recipe for high-dimensional video representations, combining Riemannian flow matching with $\boldsymbol{v}$-prediction. Systematic comparisons demonstrate significantly faster convergence than the Euclidean baseline.
- We show that the same generated world states can be rendered into multiple visual realizations or decoded into aligned depth, segmentation, optical flow and human pose, suggesting video representations as an interface for generative world modeling.

2 Related Work

Text-to-video generation. Latent diffusion models (Rombach et al., 2022) have enabled high-quality video generation in VAE space (Blattmann et al., 2023a; Yang et al., 2025; Team Wan et al., 2025; HaCohen et al., 2025; Lightricks, 2026). These methods optimize for reconstruction fidelity and generate directly in pixel-oriented latent spaces. Several concurrent efforts extend generation beyond RGB by jointly modeling multiple modalities such as depth and optical flow (Team, 2025; Chefer et al., 2025); they typically train a single model on a joint distribution of all target modalities. Our approach differs in that we generate in the representation space of a pretrained video foundation model, producing world states that already encode geometry, semantics and dynamics, and render each modality with a lightweight adapted decoder.

Generative modeling in representation space. Representation autoencoders (Zheng et al., 2025; Singh et al., 2026) demonstrate that representations from vision foundation models such as DINOv2 (Oquab et al., 2024) can serve as effective generation targets for images, offering semantic structure that accelerates training. Extending this idea to video, recent work introduces reconstruction-oriented compression to reduce the dimensionality of spatiotemporal representations before generation (Guo et al., 2026; Xie et al., 2026; Bai et al., 2025). We show that such compression weakens the semantic organization that motivates representation-space generation, and instead model the native, high-dimensional representations directly using Riemannian flow matching (Chen & Lipman, 2024; Kumar & Patel, 2026).

World models and multimodal generation. Predictive world models learn to forecast future states from actions or observations (LeCun, 2022; Bardes et al., 2024; Bruce et al., 2024). While they typically focus on deterministic prediction or planning, our work uses probabilistic flow matching to *generate* world states from text, bridging generative video synthesis and world modeling. The generated states then serve as a shared interface from which multiple modalities can be rendered.

3 Method

Existing text-to-video methods model a conditional distribution over VAE latents. The VAE's perceptual compression (Rombach et al., 2022) makes flow-based modeling practical, but its reconstruction objective biases the latents toward appearance, offering limited support for spatiotemporal structure and semantics not prominent in RGB.

In this work, we propose ***WorldFlow***, a novel paradigm that go beyond pixels: we model the *underlying world states* of a video as a sequence of compact representations containing spatial structure, temporal dynamics, and rich semantics, before rendering out their visual *appearance*. Let $\boldsymbol{c}$ denote the text condition, $\boldsymbol{s}$ the world states, and $\boldsymbol{y}$ the VAE latents of the corresponding video, we model their conditional joint distribution as

$$p_{\theta,\phi}(\boldsymbol{s}, \boldsymbol{y} \mid \boldsymbol{c}) = p_\theta(\boldsymbol{s} \mid \boldsymbol{c})\, p_\phi(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{c}), \tag{1}$$

where θ and ϕ parameterize the state generator and the generative visual renderer, respectively. As shown in Fig. 2, the generation process is naturally disentangled into two stages: the first stage performs flow matching directly in native V-JEPA representation space, generating latent world states conditioned on text prompt; the second stage then conduct flow matching in VAE space, rendering visual outputs conditioned on both text and the generated world states.

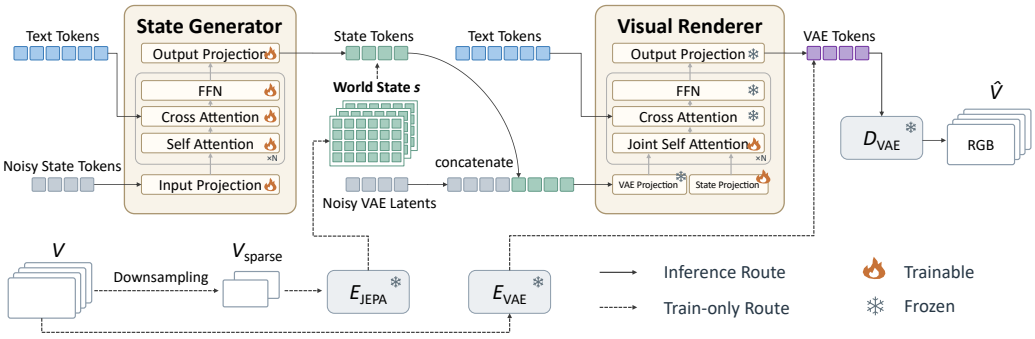

Figure 2: **WorldFlow architecture.** The state generator produces sparse latent world states in native V-JEPA 2.1 representation space, which the visual renderer uses to generate VAE latents for RGB decoding. Text conditions both stages through cross-attention, while projected state tokens and noisy VAE tokens interact through joint self-attention in the renderer. During training, frozen encoders provide state and VAE targets; encoded states also serve as renderer conditions. We train the full state generator and only the renderer's state projection and joint self-attention layers.

We describe our choice of modeling space in Sec. 3.1, the two-stage design in Sec. 3.2, the flow-matching recipe for high-dimensional native representations in Sec. 3.3, and the rendering stage and its extensions in Sec. 3.4.

3.1 Modeling Space

An ideal generation space for world states should capture scene geometry, dynamics and physical properties beyond appearance. Inspired by representation autoencoders (Zheng et al., 2025), we build on V-JEPA 2.1 (Mur-Labadia et al., 2026), whose self-supervised features support prediction, planning, and dense understanding. Its predictive objective, however, deliberately discards pixel detail deemed unpredictable or irrelevant (LeCun, 2022; Bardes et al., 2024). Generating these representations and decoding them directly to RGB is therefore suboptimal: the appearance details needed for faithful decoding are not fully determined by them. This motivates our two-stage design.

VFM representations and VAE latents are complementary. As shown in Fig. 5, the patch-wise similarity patterns in native V-JEPA 2.1 representations follow object-level semantic relationships, while those in LTX VAE follow low-level cues such as color and texture. This reflects their different training objectives: foundation models learn global and local structures through masked modeling, while VAEs are optimized for pixel-level reconstruction. We therefore assign each space the role it is suited to: modeling rich world states in representation space and synthesizing visual appearance in VAE space.

VFM representations should not be compressed. A natural concern is that high-dimensional representations are hard to model, motivating compression (Guo et al., 2026; Xie et al., 2026; Bai et al., 2025). However, as Fig. 5 shows, compressed V-JEPA patterns shift toward those of VAE latents, weakening the semantic organization that motivates generating in representation space. This gap widens in dynamic scenes (Fig. 6 in the Appendix), where native V-JEPA tracks the foreground object across frames while compressed V-JEPA and VAE latents confuse it with texture-similar background.

We therefore keep the *native* output V-JEPA 2.1 representation as the first stage generation target, with no additional compression bottleneck, and address the resulting high dimensionality through input downsampling and the flow-matching recipe described in Sec. 3.3 instead.

3.2 WorldFlow Framework

Modeling targets. Let E_{JEPA} denote the V-JEPA 2.1 encoder and $E_{\text{VAE}}, D_{\text{VAE}}$ the LTX VAE encoder and decoder, all frozen throughout training. Given a video $V \in \mathbb{R}^{T\times H\times W\times 3}$, we define the

generation targets as

$$\boldsymbol{s} = E_{\text{JEPA}}(V_{\text{sparse}}), \qquad \boldsymbol{y} = E_{\text{VAE}}(V), \qquad \hat{V} = D_{\text{VAE}}(\hat{\boldsymbol{y}}). \tag{2}$$

For a video V at 24 fps, we apply 2×2 spatial resizing and keep one frame in four, giving $V_{\text{sparse}} \in \mathbb{R}^{T/4 \times H/2 \times W/2 \times 3}$ at 6 fps, which both matches the V-JEPA pretraining setting and reduces dimensionality. The final-layer tokens $\boldsymbol{s}$ of size $\frac{T}{8} \times \frac{H}{32} \times \frac{W}{32} \times 1664$ are the stage-one target, taken without pooling or compression. Stage two targets the LTX VAE latents $\boldsymbol{y}$ of size $\frac{T}{8} \times \frac{H}{32} \times \frac{W}{32} \times 128$ encoded from the full-resolution video; the generated $\hat{\boldsymbol{y}}$ is decoded into $\hat{V}$.

Network architecture. We model both stages using the Diffusion Transformer (DiT) architecture and flow matching objective (Fig. 2). We adopt LTX-2.3 (HaCohen et al., 2025; Lightricks, 2026) as the base model for both the state generator p_θ and the renderer p_ϕ, with specific designs. (1) For stage one, the LTX backbone is adopted with only the input and output projections replaced to match the channel dimension of $\boldsymbol{s}$. The text prompt, encoded by a frozen Gemma3-12B (Gemma Team, 2025), is injected through cross-attention. All parameters along the video-generation path are trained. (2) For stage two, state tokens are linearly projected to the backbone width and concatenated with the noisy VAE tokens into a single self-attention sequence; state tokens carry timestep zero to mark them as clean. Text enters through cross-attention. During training the condition is the encoded state $\boldsymbol{s}$; at inference it is the stage-one output $\hat{\boldsymbol{s}}$. Only the state projector and self-attention layers are trained; the audio route is disabled for both stages.

Both stages are separately fine-tuned from pretrained weights rather than trained from scratch. Training and inference designs for each stage are given in Sec. 3.3 and Sec. 3.4, respectively; experiment details and full training hyperparameters are listed in Sec. 4.1.

3.3 Flow Matching in Native Representation Space

Modeling native video representations is non-trivial. Even with input downsampling, a 5-second 480p clip yields over 5k state tokens of $1{,}664$ channels, an effective data dimension (Zheng et al., 2025) near $9{\times}10^6$. Training with the standard flow matching objective converges poorly at this scale (Fig. 3), motivating us to seek a better recipe.

Representation geometry. Prior work observes that LayerNorm concentrates image representations such as DINOv2 on a hypersphere of nearly fixed radius, creating a hard shell geometry (Kumar & Patel, 2026). V-JEPA 2.1 exhibits the same structure: its non-affine LayerNorm produces tokens with zero channel mean and norm close to $\sqrt{C}$, where $C = 1664$. Each token $\boldsymbol{s}_i$ of $\boldsymbol{s}$ in equation 2 therefore lies on a centered hypersphere $\mathcal{M}_R$:

$$\boldsymbol{s}_i \in \mathcal{M}_R = \left\{ \mathbf{z} \in \mathbb{R}^C \;\middle|\; \mathbf{1}^\top \mathbf{z} = 0,\ \|\mathbf{z}\|_2 = R \right\}, \qquad R = \sqrt{C}, \tag{3}$$

where $\mathbf{1}$ is the all-ones vector, so all information is carried by the angular direction $\hat{\boldsymbol{s}}_i = \boldsymbol{s}_i / R$. Standard flow matching is a geometric mismatch here: its Gaussian source retains radial variation, and its linear interpolation leaves the manifold. This motivates Riemannian flow matching on these tokenwise hyperspheres (Chen & Lipman, 2024; Kumar & Patel, 2026), which constructs probability paths intrinsically on the manifold.

Prediction parameterization. The prediction target also matters at high dimension (Li & He, 2026): under the manifold assumption, clean data lies on a low-dimensional manifold while noise and velocity are inherently off-manifold, so $\boldsymbol{x}$-prediction places lower demands on the network than $\boldsymbol{v}$-prediction. Crossing the two factors gives four settings, Euclidean or spherical geometry with $\boldsymbol{v}$- or $\boldsymbol{x}$-prediction, which we now investigate systematically.

Toy experiment. Following the protocol of Li & He (2026), we first isolate the two factors where the target distribution is known. A two-dimensional distribution is embedded onto a centered hyperspherical shell in $\mathbb{R}^D$ by a fixed random projection, holding the intrinsic structure fixed while the ambient dimension D varies; the projection is unknown to the model. We train a five-layer MLP with 256-dimensional hidden units under each setting, all with the $\boldsymbol{v}$-loss so that only geometry

and prediction space differ (Li & He, 2026), and visualize samples mapped back to two dimensions (Fig. 4 in Appendix B).

The results reproduce the failure reported by Li & He (2026): under Euclidean geometry, $\boldsymbol{v}$-prediction degrades with D and fails at high dimension, while $\boldsymbol{x}$-prediction remains stable. On the hypersphere this gap disappears and $\boldsymbol{v}$-prediction is the most consistent of the four. We therefore conclude not that one prediction space is intrinsically better, but that *once the flow is constrained to the representation manifold, the choice of prediction space ceases to be critical.*

This choice is also favorable in practice. Sampling requires a velocity, so an $\boldsymbol{x}$-prediction model must convert its output at every step through a division that becomes singular as the noise level vanishes; Li & He (2026) clip this denominator to keep it finite. Predicting $\boldsymbol{v}$ needs no such conversion and supervises directly the quantity used at inference. We therefore adopt Riemannian flow matching with $\boldsymbol{v}$-prediction for stage one, and confirm in Sec. 4.2 that it is also the strongest of the four settings on real videos.

Training objective. We adopt Riemannian flow matching with $\boldsymbol{v}$-prediction on $\mathcal{M}_R$, orienting time so that $t = 0$ is data and $t = 1$ is noise. The source is uniform on $\mathcal{M}_R$, obtained by centering and normalizing an isotropic Gaussian: $\boldsymbol{s}_1 = R\,P\boldsymbol{g}/\|P\boldsymbol{g}\|$ with $\boldsymbol{g} \sim \mathcal{N}(0, I_C)$ and $P = I - \frac{1}{C}\mathbf{1}\mathbf{1}^\top$. With $\boldsymbol{l} = \log_{\boldsymbol{s}_0}(\boldsymbol{s}_1)$, $\theta = \|\boldsymbol{l}\|/R$ the geodesic angle, and $\mathrm{sinc}(a) = \sin(a)/a$, the constant-speed geodesic path and its velocity are

$$\boldsymbol{s}_t = \exp_{\boldsymbol{s}_0}(t\boldsymbol{l}) = \cos(t\theta)\,\boldsymbol{s}_0 + t\,\mathrm{sinc}(t\theta)\,\boldsymbol{l}, \qquad \boldsymbol{v}_t = \dot{\boldsymbol{s}}_t = -\theta\sin(t\theta)\,\boldsymbol{s}_0 + \cos(t\theta)\,\boldsymbol{l}, \tag{4}$$

which stays on $\mathcal{M}_R$ with velocity everywhere tangent, so the network output is projected onto $T_{\boldsymbol{s}_t}\mathcal{M}_R$ before the loss:

$$\mathcal{L}_\theta = \mathbb{E}_{t,\boldsymbol{s}_0,\boldsymbol{s}_1}\left[\frac{1}{NC}\sum_{i=1}^{N} w_i \left\|\Pi_{\boldsymbol{s}_{t,i}}\big(v_\theta(\boldsymbol{s}_t, t, \boldsymbol{c})_i\big) - \boldsymbol{v}_{t,i}\right\|^2\right], \qquad w_i = \mathrm{sinc}^2\big((1-t)\theta_i\big), \tag{5}$$

where $\Pi_{\boldsymbol{s}_t}$ is the tangent projection, N the number of state tokens, and $w_i = \mathrm{sinc}^2\big((1-t)\theta_i\big)$ the Jacobi regularization weight of Kumar & Patel (2026), adopted unchanged: since the exponential map on a positively curved manifold is not an isometry, it rescales each token's tangent residual so that the loss reflects endpoint error.

Timestep schedule. The budget should be spent where a single step destroys the most signal, which grows with target dimension (Esser et al., 2024). We therefore shift the timestep distribution toward higher noise in proportion to the effective data dimension NC (Zheng et al., 2025):

$$t = \sigma(\mu + g), \qquad g \sim \mathcal{N}(0,1), \qquad \mu = \tfrac{1}{2}\ln\big(NC/d_{\mathrm{ref}}\big), \tag{6}$$

with $\sigma(\cdot)$ the logistic function and $d_{\mathrm{ref}} = 4096$ the backbone reference dimension, giving $\mu \approx 3.84$ for our target.

Sampling and guidance. At inference we integrate along $\mathcal{M}_R$ with $K = 30$ Euler steps in the exponential map. Since all guidance branches at a given $\boldsymbol{s}_t$ predict into the same tangent space, they combine linearly without leaving the manifold. We use classifier-free guidance (Ho & Salimans, 2021) against a fixed negative prompt $\boldsymbol{c}_{\mathrm{neg}}$, together with spatiotemporal guidance (Hyung et al., 2025), which contrasts the model against a weakened copy $\tilde{v}_\theta$ of itself: The two guidance terms are combined in the tangent space and the state is updated by the exponential map; we give the full expression in Appendix A. The weakened branch replaces the attention output of a single mid-network block with a value bypass. Each step is followed by re-centering and re-normalization, keeping the trajectory on $\mathcal{M}_R$.

3.4 Conditional Rendering in VAE Space

Conditional flow matching in VAE space. The renderer models $p_\phi(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{c})$ by standard Euclidean flow matching. Unlike the state generator, VAE latents carry no manifold constraint, so we use a linear probability path with a Gaussian source and an unweighted objective:

$$\boldsymbol{y}_t = (1-t)\,\boldsymbol{y} + t\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \qquad \mathcal{L}_\phi = \mathbb{E}\big[\|v_\phi(\boldsymbol{y}_t, t, \boldsymbol{s}, \boldsymbol{c}) - (\boldsymbol{\epsilon} - \boldsymbol{y})\|^2\big]. \tag{7}$$

Table 1: **Text-to-video generation comparisons.** We compare WorldFlow with pretrained LTX-2.3 and its continued-training (CT) baseline, fine-tuned on the same data for 100k steps. FVD is computed on our held-out validation set with CFG disabled ($w_{\text{cfg}} = 1$); the remaining columns follow the VBench protocol with each method's selected CFG settings. Higher is better except for FVD. Best results are shown in bold.

Method	FVD ↓	Temporal Consistency & Structure			Appearance		
		Subject Consist.	Background Consist.	Spatial Relation.	Aesthetic quality	Imaging quality	Color
LTX-2.3	99.37	90.13	93.45	81.69	**64.82**	**68.15**	80.33
LTX-2.3 + CT	103.25	90.76	94.90	84.03	63.54	66.24	83.50
WorldFlow (ours)	**83.57**	**92.10**	**94.94**	**85.66**	64.12	66.95	**84.41**

The timestep schedule is inherited from the LTX backbone rather than equation 6: the shift μ is interpolated linearly in the number of target tokens, which counts the video latent tokens only and excludes the state tokens appended as conditions. Sampling follows the backbone LTX solver and guidance, with the velocity converted to a denoised endpoint at each step.

State conditioning. State tokens are linearly projected to the backbone width and concatenated with the noisy video-latent tokens into one sequence, so that a single self-attention per layer covers interactions within and across the two token types (Sec. 3.2). The state tokens are assigned a timestep of zero to mark them as clean, while the video tokens carry the sampled t; only the video tokens are supervised. The two token grids use a shared RoPE parameters, aligning each token's positional encoding by its original video coordinate rather than by grid index.

Condition noise augmentation. The renderer is trained on encoded states $\boldsymbol{s} = E_{\text{JEPA}}(V_{\text{sparse}})$ but must operate on *generated* states at inference. In order to increase robustness against possible remaining noises in generated states, we add perturbation on the clean encoded states at training:

$$\tilde{\boldsymbol{s}} = \Pi_{\mathcal{M}}\big(\boldsymbol{s} + \sigma \boldsymbol{k}\big), \qquad \sigma \sim \mathcal{U}(0, \sigma_{\max}), \qquad \boldsymbol{k} \sim \mathcal{N}(0, I), \tag{8}$$

where $\Pi_{\mathcal{M}}$ re-centers and re-normalizes onto $\mathcal{M}_R$. The perturbation is applied before the state projector and σ is not exposed to the network, so the renderer must become robust to imperfect conditions rather than compensate for a known corruption level. We use $\sigma_{\max} = 0.8$, and no perturbation is applied at inference.

Beyond naive RGB rendering. Since the visual renderer is trained to condition on text prompt to add visual details on a "scaffold" scene whose geometric structure and dynamics are already determined by the world state, our framework natively supports applications beyond RGB rendering: by editing the second stage prompt alone, a single world state can be rendered into multiple visual realizations, enabling structure-preserving re-rendering and visual editing. With a lightweight adaptation on multimodal data, the renderer can also decode depth, optical flow, human pose and other physical properties already carried by the states generated in stage one. Implementation details and visual results are given in Appendix F.

4 Experiments

4.1 Experimental Setup

Implementation details. We generate 768×448 videos (5 s, 24 FPS). Both stages are initialized from LTX-2.3 (22B) (Lightricks, 2026) and trained with AdamW. Stage one trains the full video-generation path for 100k steps (batch 32); stage two trains only the self-attention blocks and the state projector for 10k steps on clean states, then continues with noise augmentation equation 8 for another 60k steps. All flow geometry is computed in FP32; the transformer runs in BF16. Sampling uses 30 Euler steps in both stages, with CFG and STG as in equation 9. Full hyperparameters are given in Sec. 4.1.

Table 2: **Ablations on inference CFG.** All FVD scores are computed with the other stage CFG disabled (CFG=1).

w_{cfg}	FVD (stage1) ↓	FVD (stage2) ↓
1	83.57	83.57
2	80.64	83.26
3	83.44	83.66
4	<u>93.89</u>	85.22

Table 3: **Ablations on noise augmentation.** Each renderer is evaluated on encoded states and on states generated by stage one.

Noise Aug.	FVD ↓	
	encoded	generated
w/o	14.59	85.36
w.	15.12	83.57

Dataset. For text-to-video generation, we construct an internal corpus of over 3M videos spanning diverse scenes and motion types, each paired with a detailed structured caption. The caption is further translated to LTX style with Qwen3.6-35B-A3B (Qwen Team, 2026) for training. Representative captions are given in the Appendix. We hold out $2{,}048$ videos disjoint from the training set as a validation set. For multimodal rendering, we use a small subset of publicly available multimodal video data for efficient adaptation.

Evaluation protocol. We report FVD on the validation set, which shares the training caption distribution and therefore measures fit to the target distribution; we use it for all ablations. We additionally report VBench (Huang et al., 2024) on its standard prompt suite, separating the structural and temporal dimensions our method should affect (subject consistency, background consistency, spatial relationship) from appearance measures (aesthetic and imaging quality) largely inherited from the renderer. Unless stated otherwise we sample with $w_{\text{cfg}} = 3$ for stage one and 4 for stage two, both with $w_{\text{stg}} = 1$, and fix the seed across methods.

Baselines. Our reference is the LTX-2.3 backbone from which both stages are initialized. Since comparing against the released checkpoint alone would confound our contribution with the effect of additional training on our corpus, we also finetune the backbone on the same data for the same number of steps as a controlled baseline. All baselines are evaluated at 100k steps with $w_{\text{cfg}} = 4$ and $w_{\text{stg}} = 1$.

4.2 Text-to-Video Generation

Quantitative results. Table 1 reports the comparison against both baselines. For FVD, we evaluate all methods with CFG disabled ($w_{\text{cfg}} = 1$) to reduce the influence of guidance. For VBench, we evaluate each method with its selected CFG setting.

WorldFlow improves structural and temporal consistency while maintaining comparable aesthetic and imaging quality. This is consistent with the affinity patterns in Fig. 6: native representations capture high-level structure and long-range relationships, which may facilitate coherent generation.

Qualitative results. Qualitative comparisons at matched prompts, seeds, and training iterations are shown in Appendix G. WorldFlow consistently reproduces the spatial layout and object identities specified by the prompt throughout the clip, while the baseline produces visually plausible frames that gradually drift from the intended structure.

Ablation studies. Table 2 compares how w_{cfg} affects FVD in each stage. Increasing guidance causes only a modest change in the renderer's FVD, whereas the state generator degrades markedly at $w_{\text{cfg}} = 4$. Although stage one achieves its lowest FVD at $w_{\text{cfg}} = 2$, we use $w_{\text{cfg}} = 3$ for stage one and $w_{\text{cfg}} = 4$ for stage two to balance text alignment and generation quality.

Table 3 examines whether condition noise augmentation improves the renderer's robustness to the gap between encoded and generated states. We compare renderers trained with and without augmentation on encoded states, which provide a reference, and on states generated by stage one, which correspond to the inference setting. Augmentation slightly increases FVD on encoded states from 14.59 to 15.12, while reducing it on generated states from 85.36 to 83.57. This trade-off supports its use for rendering imperfect state predictions.

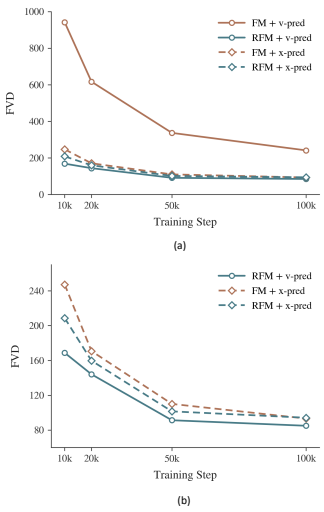

Figure 3: **Ablation study on flow geometry and prediction parameterization.** FVD (↓) of rendered videos over state-generator training, comparing Euclidean flow matching (FM) and hyperspherical Riemannian flow matching (RFM) with $\boldsymbol{v}$- or $\boldsymbol{x}$-prediction. **(a)** All four configurations. **(b)** A closer view of the three configurations excluding Euclidean $\boldsymbol{v}$-prediction. RFM with $\boldsymbol{v}$-prediction achieves the lowest FVD at every evaluated checkpoint, whereas Euclidean $\boldsymbol{v}$-prediction remains substantially worse even after 100k training steps.

Fig. 3 compares FVD over training across combinations of flow geometry and prediction parameterization. Consistent with the toy experiment (Fig. 4), Euclidean flow matching with $\boldsymbol{v}$-prediction fails to converge, whereas the other three settings are effective. Riemannian flow matching with $\boldsymbol{v}$-prediction achieves the lowest FVD at every evaluated checkpoint.

4.3 Appearance Editing and Multimodal Rendering

Since the state generator and visual renderer take separate prompts at inference, holding a generated state $\boldsymbol{s}$ fixed and varying only the rendering prompt $\boldsymbol{c}_y$ edits appearance — color, material, style, lighting — while retaining spatiotemporal organization, without additional training. The same states can also drive lightweight adapted renderers that produce depth, optical flow, segmentation, DensePose, and human pose using empty text prompts, so any structure in the outputs reflects information encoded in $\boldsymbol{s}$ alone. Full details, equations, and visual results are given in Appendix F.

5 Conclusion

We presented WorldFlow, a two-stage framework that generates latent world states in native video representation space before rendering their visual appearance. By formulating state generation as Riemannian flow matching on the representation manifold, we show that high-dimensional video representations can be modeled effectively without compression. The generated states improve structural and temporal consistency in text-to-video synthesis, and serve as a shared interface for appearance editing and multimodal rendering. We hope this work encourages further exploration of representation-space generation as a foundation for controllable and interpretable video synthesis.

AI Use Statement

In this work, we used large language models (LLMs) to assist with writing and editing portions of the manuscript text, including polishing of explanations and proofreading for clarity and grammar. We also used LLM-assisted code generation to help implement evaluation scripts and data preprocessing pipelines. All AI-assisted writing was reviewed, revised, and verified by the authors; all AI-assisted code was tested for correctness by at least one author. We did not use generative AI to produce any experimental results, figures, or claims reported in this work, and the core research ideas, method design, and analysis are entirely the authors' own. We take full responsibility for the final content of this work.

Ethics Statement

This work presents a method for text-to-video generation. The primary intended use is creative and research applications. As with all generative video models, the underlying techniques could in principle be misused to produce misleading or harmful visual content. We are aware of these concerns and believe that responsible deployment, content moderation practices, and ongoing work on detection methods are important complements to advances in generation. This work does not involve human subjects studies, personally identifiable information, or novel datasets with sensitive content. The training data used in our experiments is drawn from existing public or licensed sources; we follow the respective terms of use. We have no conflicts of interest to declare.

Reproducibility Statement

To support reproducibility, we provide the following. **Method description.** The WorldFlow framework, including both the state-generator and visual-renderer stages, is described in full in Sec. 3. The Riemannian flow matching loss, timestep schedule, and sampling procedure are given in Sec. 3.3 and Appendix A, with all hyperparameters stated explicitly. **Evaluation protocol.** Baselines, metrics, and evaluation settings are described in Sec. 4.1; the VBench evaluation follows the standard protocol and we report all dimensions used. **Code.** An anonymized implementation will be released as supplementary material. **Model checkpoints.** Both stages are initialized from the publicly available LTX-2.3 weights; fine-tuned checkpoints will be released upon acceptance.

References

Jianhong Bai, Xiaoshi Wu, Xintao Wang, Xiao Fu, Yuanxing Zhang, Qinghe Wang, Xiaoyu Shi, Menghan Xia, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Kun Gai. SemanticGen: Video generation in semantic space. *arXiv preprint arXiv:2512.20619*, 2025. URL https://arxiv.org/abs/2512.20619.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. URL https://arxiv.org/abs/2404.08471.

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. *arXiv preprint arXiv:2311.15127*, 2023a. URL https://arxiv.org/abs/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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023b. URL https://arxiv.org/abs/2304.08818.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. *arXiv preprint arXiv:2502.02492*, 2025.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2302.03660.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2403.03206.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Minghui Guo, Shengqiong Wu, and Hao Fei. V-RAE: Rethinking video latent spaces for generation. *arXiv preprint arXiv:2608.13556*, 2026. URL https://arxiv.org/abs/2608.13556.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2025. URL https://arxiv.org/abs/2501.00103.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Amandeep Kumar and Vishal M. Patel. Learning on the manifold: Unlocking standard diffusion transformers with representation encoders. *arXiv preprint arXiv:2602.10099*, 2026. URL https://arxiv.org/abs/2602.10099.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. Position paper, version 0.9.2.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026. URL https://arxiv.org/abs/2511.13720.

Lightricks. LTX-2: Open-weight video diffusion models, 2026. URL https://github.com/Lightricks/LTX-Video. Model release.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. URL https://arxiv.org/abs/2603.14482.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2112.10752.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026. URL https://arxiv.org/abs/2605.18324.

Aether Team. Aether: Geometric-aware unified world modeling. *arXiv preprint arXiv:2503.18945*, 2025.

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. URL https://arxiv.org/abs/2503.20314.

Zhihao Xie, Junfeng Wu, Xinting Hu, Junchao Huang, and Li Jiang. VideoRAE: Taming video foundation models for generative modeling via representation autoencoders. *arXiv preprint arXiv:2607.14088*, 2026. URL https://arxiv.org/abs/2607.14088.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ce31378e9f41d8907e97dab172b6c559-Paper-Conference.pdf.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.06940.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. URL https://arxiv.org/abs/2510.11690.

Appendix

A Stage-One Sampling and Guidance

WorldFlow-S1 uses classifier-free guidance (CFG) and spatiotemporal skip guidance (STG) (Hyung et al., 2025) simultaneously. At each solver step, we compute a conditional velocity $v_\theta(\boldsymbol{s}_t, t, \boldsymbol{c})$, a negative-prompt velocity $v_\theta(\boldsymbol{s}_t, t, \boldsymbol{c}_{\text{neg}})$, and a weakened velocity $\tilde{v}_\theta(\boldsymbol{s}_t, t, \boldsymbol{c})$ obtained by replacing self-attention mixing in a single mid-network transformer block with a value bypass. All three branches operate in the shared tangent space at $\boldsymbol{s}_t$; we combine them and then project onto $T_{\boldsymbol{s}_t}\mathcal{M}_R$ before taking the exponential-map step:

$$\begin{aligned}\hat{\boldsymbol{v}} &= v_\theta(\boldsymbol{s}_t, t, \boldsymbol{c}) + (w_{\text{cfg}} - 1)\big(v_\theta(\boldsymbol{s}_t, t, \boldsymbol{c}) - v_\theta(\boldsymbol{s}_t, t, \boldsymbol{c}_{\text{neg}})\big) \\ &\quad + w_{\text{stg}}\big(v_\theta(\boldsymbol{s}_t, t, \boldsymbol{c}) - \tilde{v}_\theta(\boldsymbol{s}_t, t, \boldsymbol{c})\big), \\ \boldsymbol{s}_{t+\Delta t} &= \exp_{\boldsymbol{s}_t}\big(\Delta t\, \Pi_{\boldsymbol{s}_t}(\hat{\boldsymbol{v}})\big), \qquad \Delta t < 0,\end{aligned} \tag{9}$$

where $\Pi_{\boldsymbol{s}_t}(\boldsymbol{u}) = P\boldsymbol{u} - \boldsymbol{s}_t\langle \boldsymbol{s}_t, P\boldsymbol{u}\rangle / R^2$ is the tangent projection and $P = I - \frac{1}{C}\mathbf{1}\mathbf{1}^\top$ removes the channel mean of each token. Since $\Pi_{\boldsymbol{s}_t}$ is linear, combining the branches before projecting is equivalent to projecting each branch separately and then combining, so the order does not affect the result. After each step, the state is re-centered and re-normalized to remain on $\mathcal{M}_R$ and prevent accumulated drift.

B Toy Experiment Visualization

To isolate the effect of flow geometry and prediction parameterization, we follow Li & He (2026) and construct a synthetic problem where the ground-truth distribution is known: a two-dimensional distribution embedded onto a centered hyperspherical shell in $\mathbb{R}^D$ via a fixed random projection, with D varying from low to high. The intrinsic structure is held fixed while the ambient dimension grows, so any degradation must be attributed to the modeling choice rather than data complexity. All four settings use an identical five-layer MLP with 256-dimensional hidden units and the same $\boldsymbol{v}$-loss; only the geometry (Euclidean vs. hyperspherical) and the prediction parameterization ($\boldsymbol{v}$-prediction vs. $\boldsymbol{x}$-prediction) differ.

The results confirm the finding of Li & He (2026): Euclidean $\boldsymbol{v}$-prediction fails at high dimension, while $\boldsymbol{x}$-prediction on the same geometry remains stable. On the hypersphere the gap between the two prediction spaces disappears, and $\boldsymbol{v}$-prediction is the most consistent of the four. The key takeaway is not that one prediction space is intrinsically superior, but that constraining the flow to the representation manifold largely removes sensitivity to the prediction parameterization.

C Additional Representation Analysis

Fig. 5 presents the full patch-wise feature similarity maps reported in Sec. 3.1. We place a query patch on one frame and compute cosine similarities against all patches in several adjacent frames; similarity is displayed as a heatmap. The maps complement the affinity analysis in the main text by showing a broader set of query locations and temporal offsets.

D Low-Light Multimodal Rendering

The low-light experiment tests whether a generated world state retains structural information even when the RGB channel carries little luminance. We supply a low-light text prompt to both the state generator and the RGB renderer, obtaining an RGB video that is nearly dark. The non-RGB renderers (depth, optical flow, DensePose, human pose) receive only an empty prompt and condition solely on the generated state.

This result illustrates a key property of the two-stage design: because the state generator operates in representation space rather than pixel space, the semantic and geometric content of a scene is encoded in $\boldsymbol{s}$ regardless of how that content manifests visually. A separate RGB renderer then interprets the same state under different appearance conditions, while non-RGB renderers access the underlying structure directly.

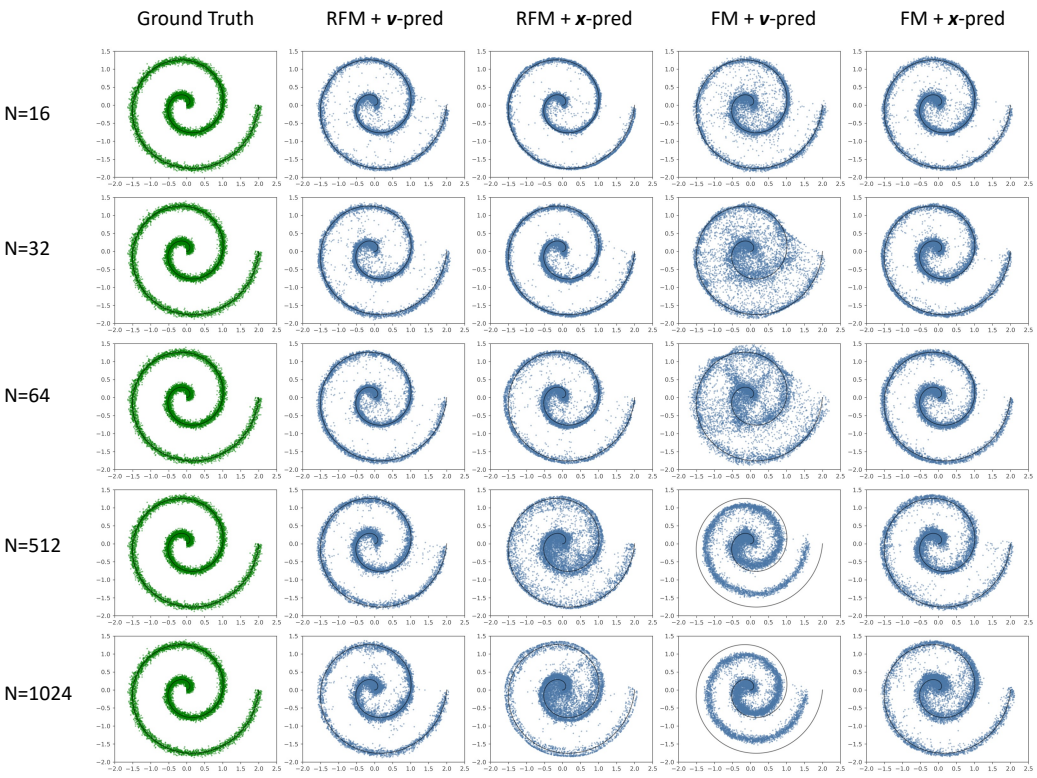

Figure 4: **Toy experiment: flow geometry and prediction parameterization at varying ambient dimension.** A two-dimensional distribution is embedded onto a centered hyperspherical shell in D-dimensional space and modeled under four flow matching formulations. Samples are projected back to two dimensions for visualization. Euclidean flow matching with $\boldsymbol{v}$-prediction degrades as D grows and fails to recover the distribution at high dimension. The remaining three settings remain effective, with Riemannian flow matching with $\boldsymbol{v}$-prediction being the most consistent across dimensions.

E Additional Multimodal Rendering Examples

Figures 8–10 show additional examples rendered in six modalities from a shared generated world state for each example. The non-RGB renderers are adapted with modality-specific supervision and use empty text prompts at inference, with the shared state providing the only scene-specific condition. The RGB renderer also receives the text prompt. The structural alignment across modalities is consistent across prompts involving human motion, rigid objects, and outdoor scenes.

F Appearance Editing and Multimodal Rendering

Appearance Editing

Since the state generator and visual renderer take separate prompts at inference, we denote the state-generation prompt by $\boldsymbol{c}_s$ and the rendering prompt by $\boldsymbol{c}_y$:

$$\boldsymbol{s} \sim p_\theta(\boldsymbol{s} \mid \boldsymbol{c}_s), \qquad \boldsymbol{y} \sim p_\phi(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{c}_y). \tag{10}$$

For standard generation, $\boldsymbol{c}_y = \boldsymbol{c}_s$. Since the renderer uses text to synthesize details not fully specified by $\boldsymbol{s}$, holding the generated state fixed and varying only $\boldsymbol{c}_y$ edits the scene's appearance — color, material, style, lighting — while retaining its spatiotemporal organization, without additional training (Fig. 11).



Figure 5: **Patch-wise Feature Similarity Analysis in Different Spaces**. For each static frame, we sample a query point and compute cosine similarities between the corresponding query patch and other patches in each feature space. V-JEPA 2.1 space exhibits clear clustering around object-level semantic entities, while LTX VAE focuses on color, texture, and local patterns. After a simple compression operation optimized with a reconstruction objective (Guo et al., 2026), however, the compressed V-JEPA 2.1 latent exhibits patterns closer to LTX VAE, suggesting an undesirable shift away from its original semantic organization.

Multimodal Rendering

The same generated world states can be rendered into additional visual modalities through lightweight adaptation:

$$\boldsymbol{y}_m \sim p_{\phi_m}(\boldsymbol{y}_m \mid \boldsymbol{s}). \tag{11}$$

where m indexes the output modality and p_{ϕ_m} the corresponding adapted renderer. We consider depth, object segmentation, optical flow, DensePose, and human pose. These renderers *use empty prompts*, with $\boldsymbol{s}$ providing the only scene-specific condition, so their outputs probe the geometry and dynamics encoded within it. Rather than training a DPT decoder, we reuse the rendering DiT to model uncertainty when rendering temporally dense outputs from sparse states; 1k adaptation iterations at batch size 16 suffice.

As shown in Fig. 12, outputs rendered from the same state are spatiotemporally aligned with the RGB video. Further more, in an extreme scenario (Fig. 7), a low-light prompt is given only to the state generator and RGB renderer; although the RGB video is nearly dark, depth, optical flow, DensePose and human pose remain clearly structured. Through a different lens, the same world state reveals scene structure barely visible in RGB.

G Additional Qualitative Results

Fig. 13 provides additional frame comparisons between LTX-2.3 and WorldFlow at matched prompts, seeds, and training iterations, extending the qualitative discussion in Sec. 4.2. In all examples the baseline receives identical conditioning; differences in structural coherence arise from the two-stage state-then-render design rather than from any difference in data or compute.

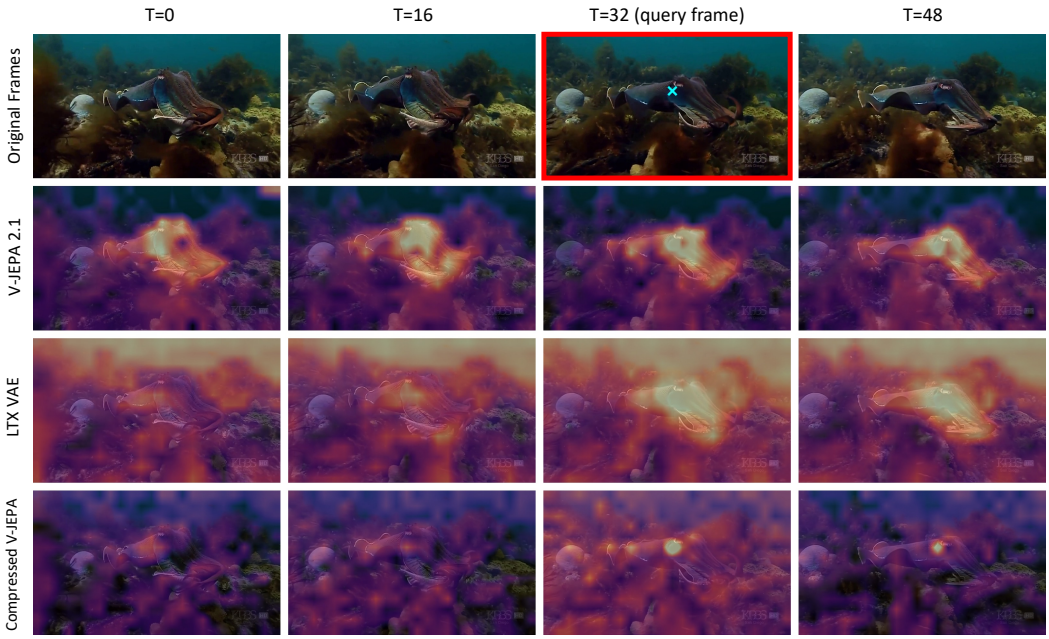

Figure 6: **Patch-wise feature similarity across frames.** A query patch is placed on one frame and similarities are computed against patches in multiple adjacent frames. Native V-JEPA 2.1 clusters patches belonging to the same object across frames, exhibiting long-horizon consistency. In contrast, LTX VAE and compressed V-JEPA 2.1 features fail to isolate the object from background regions with similar color patterns.

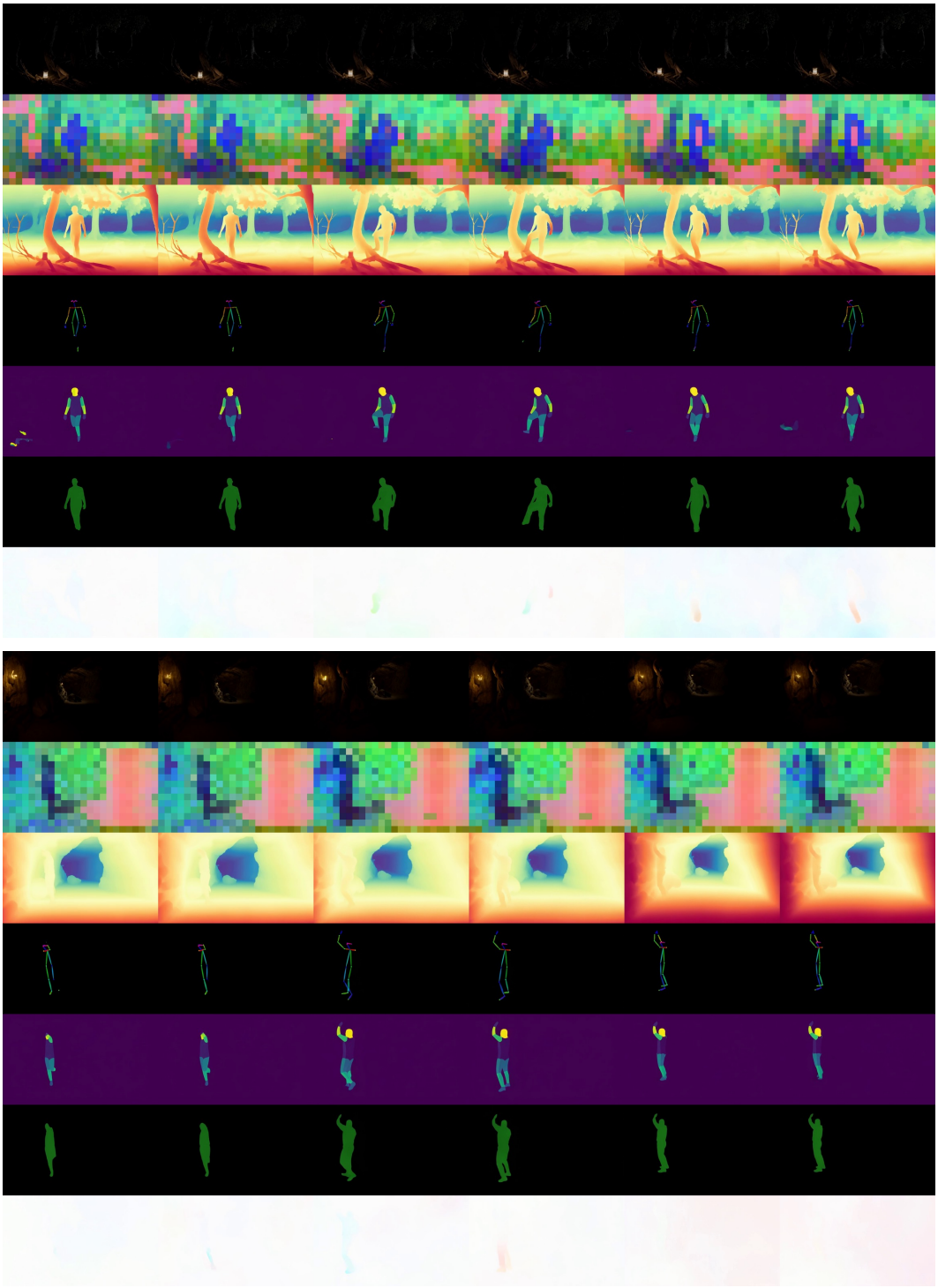

Figure 7: **Low-light multimodal rendering.** Top: RGB frames from a low-light prompt, appearing nearly dark. Bottom rows: depth, optical flow, DensePose, and human pose rendered from the same world state with empty text prompts. Despite the RGB channel carrying almost no luminance signal, the non-RGB outputs remain clearly structured, demonstrating that the world state encodes scene geometry and dynamics independently of illumination appearance.

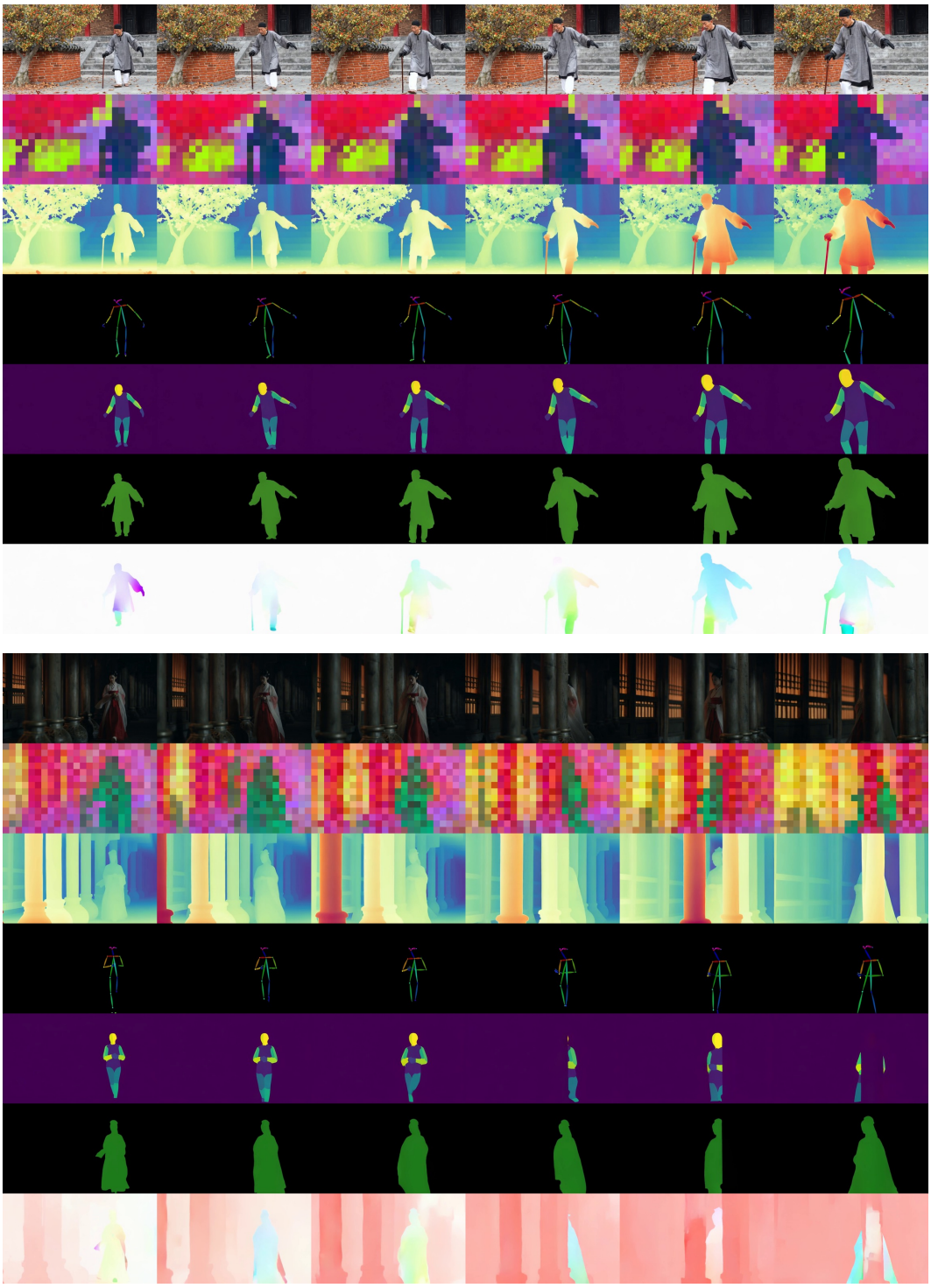

Figure 8: **More multimodal rendering results.** Rows within each example: RGB, world-state visualization, depth, human pose, DensePose, object segmentation, and optical flow. Columns show successive video frames. All rendered modalities share the same generated world state; only the non-RGB renderers use empty text prompts.

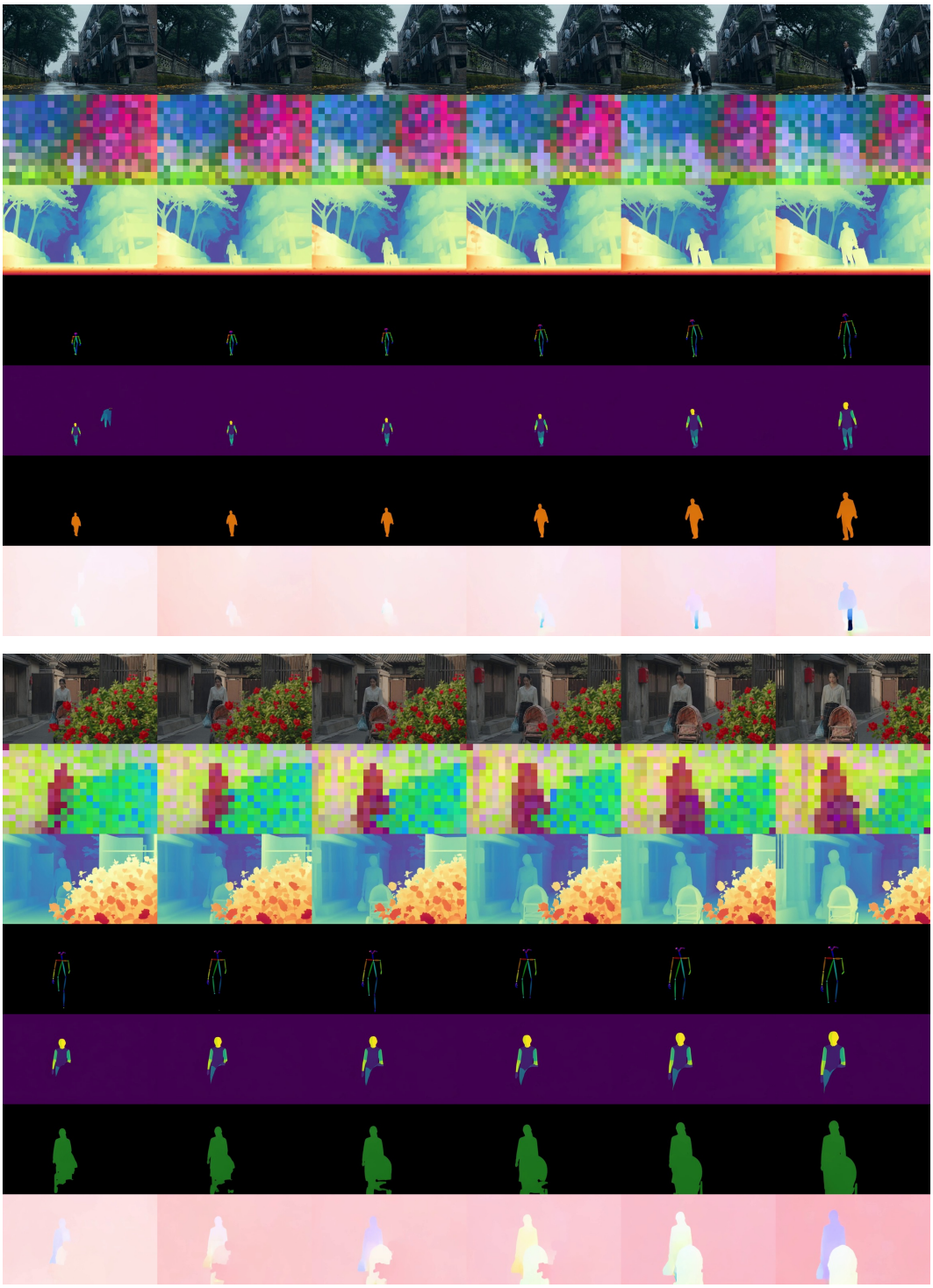

Figure 9: **Multimodal rendering results.**

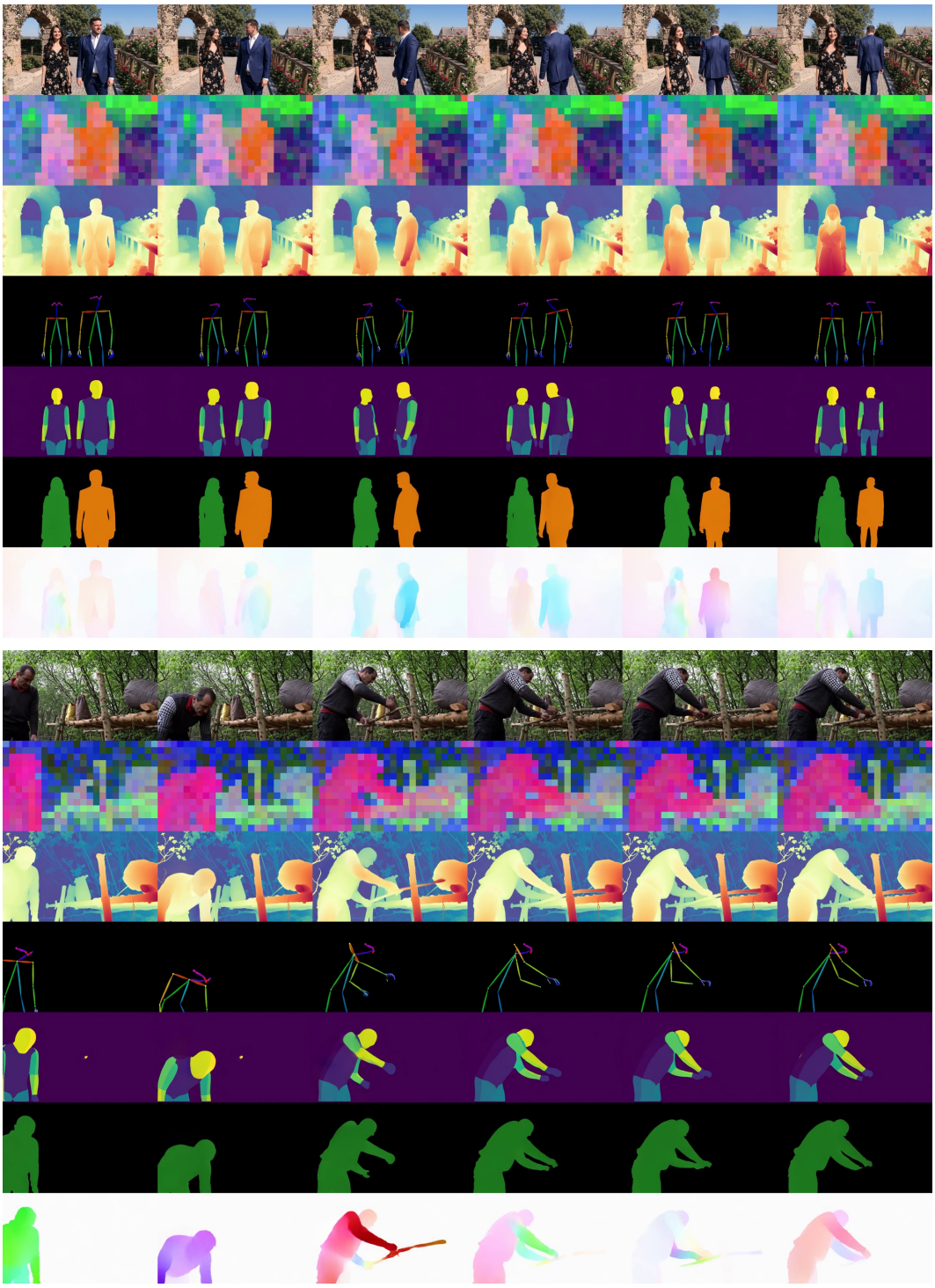

Figure 10: **Multimodal rendering results.**

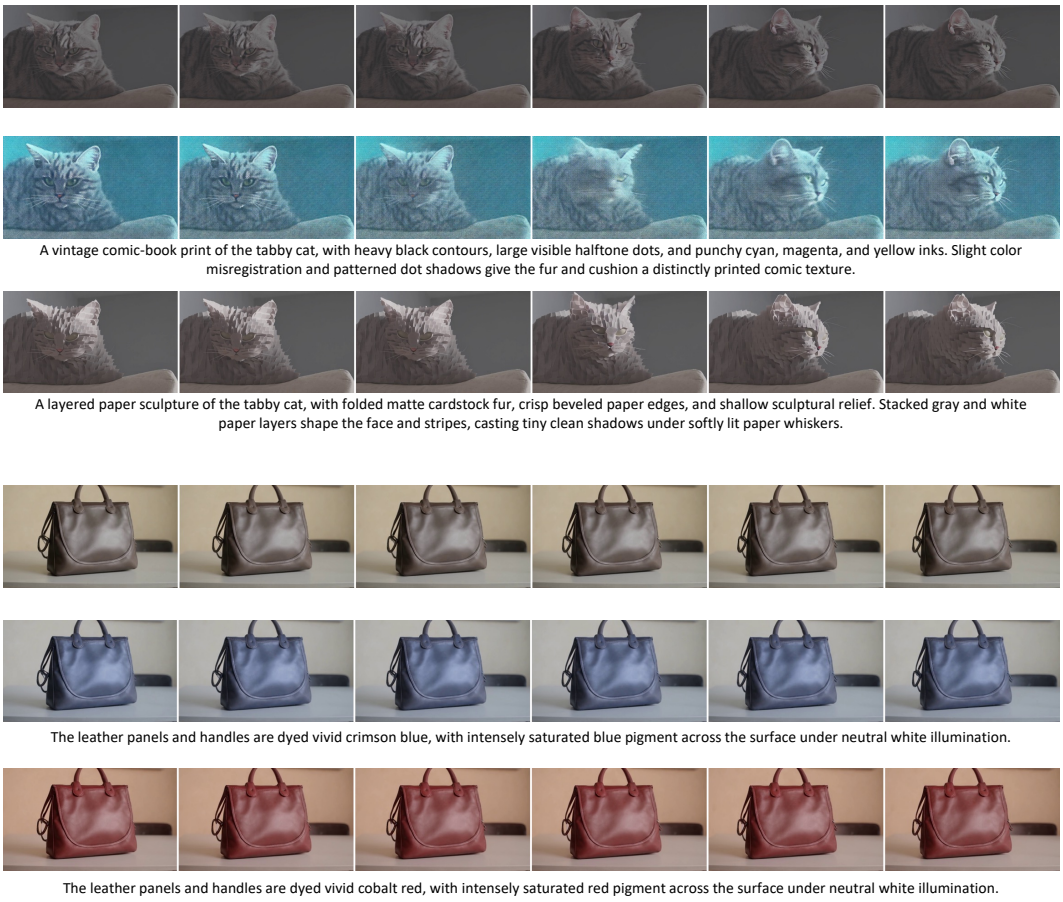

Figure 11: **Re-rendering and Appearance Editing.** With the generated state $\boldsymbol{s}$ held fixed, varying only the rendering prompt $\boldsymbol{c}_y$ produces different appearances without additional training. For each example, the first row shows the original rendering and the following rows show edited results. As shown in the exmaples, style and material changes are applied to a tabby cat and color changes to a leather handbag, while retaining the overall scene layout and motion.

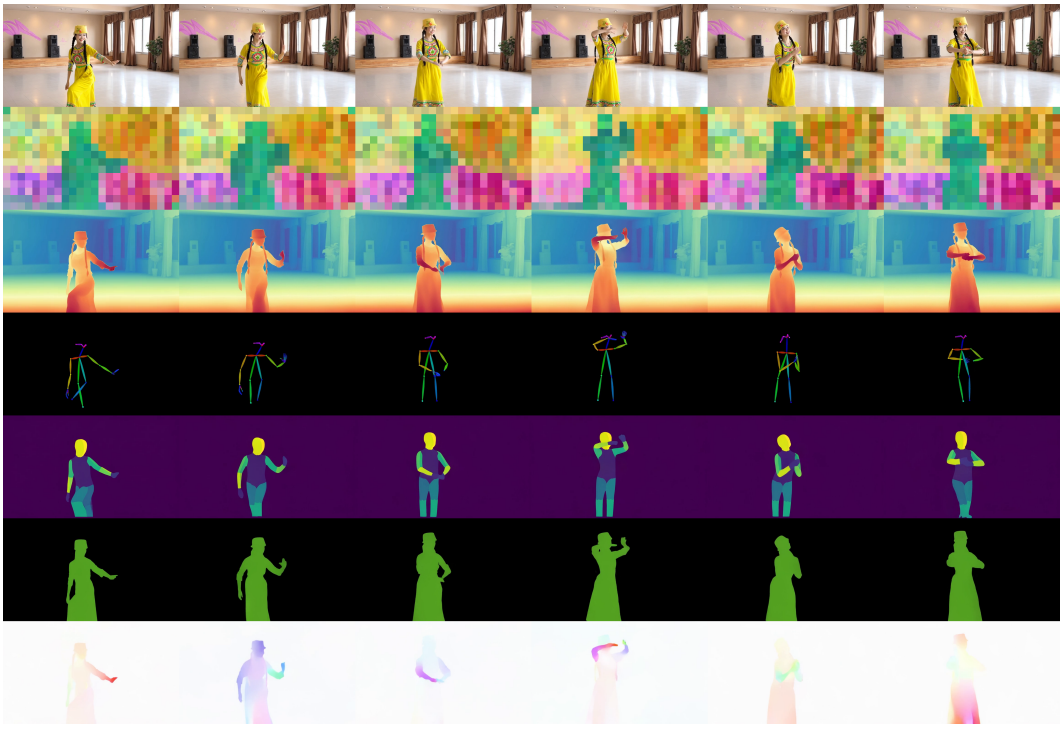

Figure 12: **Multimodal rendering from a shared world state.** Starting from the same generated state $\boldsymbol{s}$, adapted renderers produce RGB video, depth, optical flow, object segmentation, DensePose, and human pose, with spatiotemporally aligned outputs. Non-RGB renderers use *empty text prompts*; the state alone provides the scene-specific condition.

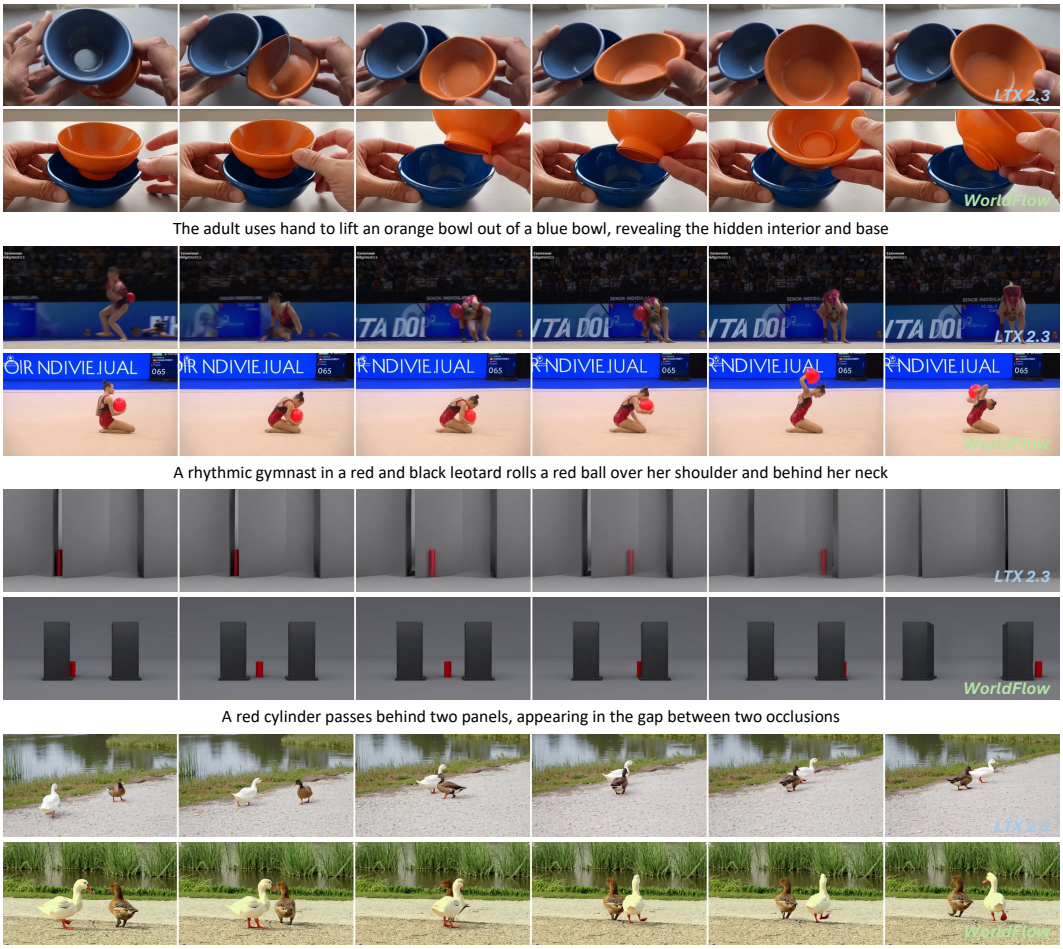

Figure 13: **Additional qualitative comparison with LTX-2.3.** For each prompt, selected frames from LTX-2.3 (top rows) and WorldFlow (bottom rows) are shown in chronological order. Prompts involve nested-object manipulation, human–object interaction, and occlusion scenarios that require consistent spatial reasoning across time. WorldFlow better preserves object geometry and the spatial relations specified by the prompts, including the bowls' nesting order, the ball's shape during contact, and the visibility of objects as they pass behind panels or one another.