FLOWLONG: INFERENCE-TIME LONG VIDEO GENERATION VIA MANIFOLD-CONSTRAINED TWEEDIE MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Extending the generation horizon of video diffusion models to long sequences remains a long-standing and important challenge. Existing training-free approaches either adapt bidirectional models through architecture-specific modifications that degrade over long horizons, or rely on autoregressive generation that accumulates drift through exposure bias and produces repetitive motion. We instead formulate long video generation as an *inverse problem*: a long video is decomposed into a chain of short chunks, each constrained to be consistent with its neighbor's denoised estimate, which serves as the measurement. Following diffusion inverse solvers, we address this problem with a *manifold-constrained* update on the denoised estimates, which we call *Tweedie matching*; it reduces to a closed-form per-frame solution and enforces temporal consistency while restricting the correction to the clean data manifold of the pretrained model. Because deterministic ODE trajectories tend to revert to their independent paths, we further propose *stochastic early-phase sampling*, which injects fresh noise after each correction in the high-noise phase to synchronize the windows, before switching to deterministic sampling to preserve fine-grained visual fidelity. The resulting framework is training-free and architecture-agnostic. Applied to various video generation models, it generates videos several times longer than the native window while outperforming both training-free and autoregressive baselines on VBench and long-range drift metrics, and extends without fine-tuning to audio-video joint generation and text-to-3DGS. Videos are available at `https://flowlongvideo.github.io/`.

1 INTRODUCTION

Video Diffusion Transformers (DiTs) (Peebles & Xie, 2022) have driven remarkable progress in video generation and its extensions, including camera-controlled generation (Yu et al., 2025; Bai et al., 2025; Jeong et al., 2025; Hong et al., 2025) and 3D/4D generation (Voleti et al., 2024; Go et al., 2026; Wu et al., 2025; Wang et al., 2025; Park et al., 2025). Among these, long video generation is particularly pressing for applications ranging from cinematic content creation to embodied world models (Ha & Schmidhuber, 2018; Bruce et al., 2024; Team et al., 2026; Seo et al., 2026).

Despite these demands, generating videos significantly longer than the training length remains a fundamental challenge. Most video diffusion models are trained on short clips due to the scarcity of large-scale, high-quality long video data, and directly applying them beyond their training length leads to severe quality degradation. This has motivated a growing body of work on long video generation, which falls into two categories. The first extends pre-trained bidirectional video diffusion models to longer sequences without additional training (e.g., FIFO-Diffusion (Kim et al., 2024b), RIFLEx (Zhao et al., 2025a), UltraViCo (Zhao et al., 2025b)). While these methods avoid additional training, they share limitations: consistency degrades as video length grows, visual artifacts accumulate over long horizons, and their reliance on architecture-specific modifications hinders applicability to new models. The second category formulates long video generation autoregressively. CausVid (Yin et al., 2025) demonstrates that distillation-based few-step generation can be applied to video, enabling autoregressive generation via KV-cache. Self-Forcing (Huang et al., 2025) further addresses the training-inference gap, inspiring follow-up works (Cui et al., 2025; Liu et al., 2025; Yi et al., 2025; Yesiltepe et al., 2025). However, these methods suffer from several limitations. Reusing KV-cache

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.

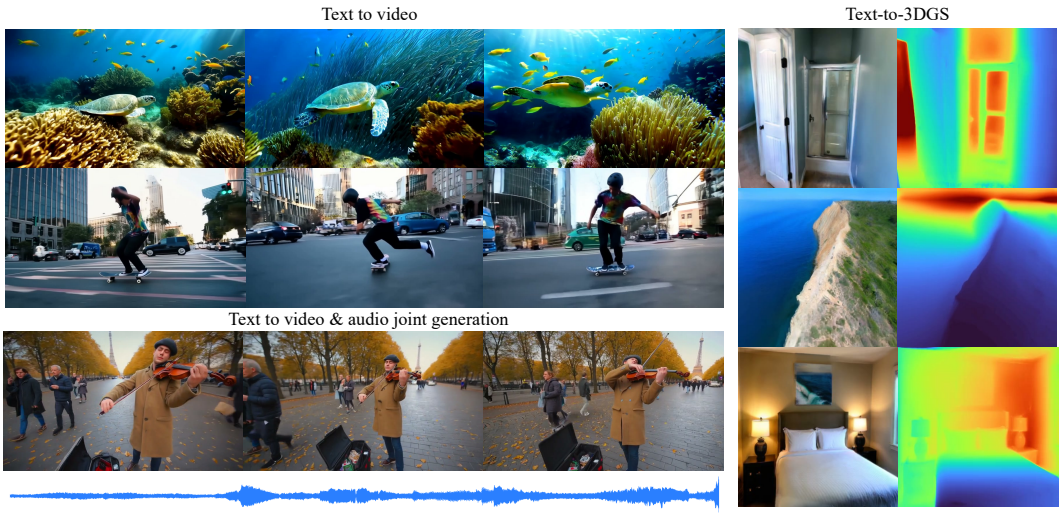


Figure 1: **Qualitative results.** *FlowLong* extends pretrained video diffusion models beyond their native context window without any additional training. Being model-agnostic and training-free, it readily applies across diverse tasks: text-to-video generation, joint audio-video generation, and text-to-3DGS. This points toward a versatile, plug-and-play framework for extrapolating the generative horizon of any flow-based model. Please refer to the supplementary video.

across segments causes errors to accumulate over time, leading to *exposure bias* and temporal drift. Motion diversity also degrades, as the model tends to produce repetitive motion patterns over long horizons. Furthermore, these approaches require distillation from a bidirectional teacher model, making them difficult to apply on the fly to recently introduced architectures such as joint audio-video models (HaCohen et al., 2026).

To overcome these limitations, we propose a novel inference-time framework for long video generation, grounded in the geometric view of flow-based video generative models. Inspired by recent advances in diffusion-based inverse problem solvers (Chung et al., 2023), we reformulate long video generation as an inverse problem that aligns multiple chunk sampling trajectories toward a coherent sequence. Specifically, we regularize each chunk's denoising path with a guidance loss enforcing smooth and manifold-constrained transitions across shared frames of adjacent chunks. This reduces to a one-step gradient correction on the denoised estimate during reverse sampling, which takes the closed-form of a simple per-frame interpolation on the overlap region—a procedure we call *Tweedie matching*. To sustain the effect of this correction and prevent trajectories from reverting to their divergent ODE paths, we further propose *stochastic early-phase sampling*: noise is injected during the initial stages to break ODE trajectory inertia so that Tweedie matching can synchronize adjacent chunks, before transitioning to deterministic ODE sampling phase. Since our framework modulates only the sampling process, it is fully architecture-agnostic, training-free, and avoid the exposure bias inherent in KV-cache reuse. It further extends seamlessly to audio-video joint generation and text-to-3D scene generation without any fine-tuning. Our contributions are as follows:

- We propose **FlowLong**, a training-free, model-agnostic framework that extends pretrained flow-based diffusion models beyond their native generation horizon. Operating purely at inference time, FlowLong applies uniformly to text-to-video, audio-video joint, and text-to-3D scene generation without any architectural modification or fine-tuning.
- To our knowledge, we are the first to formulate long video generation as an inverse problem, from which we derive *Tweedie matching*, which matches the denoised estimates of adjacent chunks under the manifold constraint, and *stochastic early-phase sampling*, which breaks ODE trajectory inertia in the high-noise regime.
- We validate FlowLong across text-to-video, audio-video joint generation, and text-to-3DGS, consistently outperforming both training-free and autoregressive baselines in qualitative and quantitative evaluations without any fine-tuning or backbone-specific modifications.

2 Related Work

Bidirectional video diffusion. Recent video diffusion models (Wan et al., 2025; Kong et al., 2024; HaCohen et al., 2026) generate a fixed-length window through full spatio-temporal attention. Training-free extensions rely on backbone-specific interventions: FIFO-Diffusion (Kim et al., 2024b) denoises a queue with monotonically increasing noise levels, while RIFLEx (Zhao et al., 2025a) and UltraViCo (Zhao et al., 2025b) modify positional frequencies and attention scores, respectively. These interventions couple each method to particular architectures, and quality still degrades beyond the training length. FlowLong instead leaves the backbone untouched and harmonizes overlapping windows, decoupling video length from the native window size.

Autoregressive video diffusion. Autoregressive approaches (Chen et al., 2024; Liu et al., 2025; Yin et al., 2025; Huang et al., 2025; Yi et al., 2025) enable fast generation by extending distribution matching distillation (Yin et al., 2024) to videos. Follow-up works mitigate error accumulation over long horizons via rolling KV caches (Cui et al., 2025; Liu et al., 2025), context compression (Zhang et al., 2025a), or learned history encoders (Zhang et al., 2025b). Nonetheless, conditioning on self-generated history leaves them susceptible to exposure bias, drift, and motion repetition; they require distillation or finetuning for each backbone. In contrast, FlowLong samples all windows in parallel from independent noise without KV-cache, avoiding exposure bias by construction and applying directly to new architectures such as audio-video joint models (HaCohen et al., 2026) and text-to-3DGS models (Go et al., 2026).

3 Preliminaries

Flow model. Flow matching (Liu et al., 2022) defines a continuous normalizing flow that transports samples from a simple source distribution p_1 to a target distribution p_0 over $\mathbb{R}^d$ along a straight path. For example, Rectified flow (Liu et al., 2022) defines a linear interpolant between a data sample $\boldsymbol{x}_0 \sim p_0$ and noise $\boldsymbol{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1. \tag{1}$$

A neural network $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ is trained to approximate the velocity field $\boldsymbol{v}(\boldsymbol{x}_t) = \frac{d\boldsymbol{x}_t}{dt}$ that transports $\mathbf{x}_1$ back to $\mathbf{x}_0$, via the following conditional flow matching objective:

$$\mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_1}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t,t) - \boldsymbol{v}(\boldsymbol{x}_t|\boldsymbol{x}_0)\|^2\right], \quad \boldsymbol{v}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \boldsymbol{x}_1 - \boldsymbol{x}_0 \tag{2}$$

Sampling. Starting from $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, samples ($\boldsymbol{x}_0 \sim p_0$) are generated by solving the learned ODE from $t = 1$ to $t = 0$:

$$d\boldsymbol{x}_t = v_\theta(\boldsymbol{x}_t, t)\, dt. \tag{3}$$

For example, an Euler step from time t to $s < t$ reads

$$\boldsymbol{x}_s = \boldsymbol{x}_t + (s-t)\boldsymbol{v}_\theta(\boldsymbol{x}_t, t). \tag{4}$$

Defining the denoised and noisy estimates as

$$\hat{\boldsymbol{x}}_{0|t} := \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] = \boldsymbol{x}_t - t\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) \tag{5}$$

$$\hat{\boldsymbol{x}}_{1|t} := \mathbb{E}[\boldsymbol{x}_1|\boldsymbol{x}_t] = \boldsymbol{x}_t + (1-t)\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) = \frac{\boldsymbol{x}_t - (1-t)\hat{\boldsymbol{x}}_{0|t}}{t}, \tag{6}$$

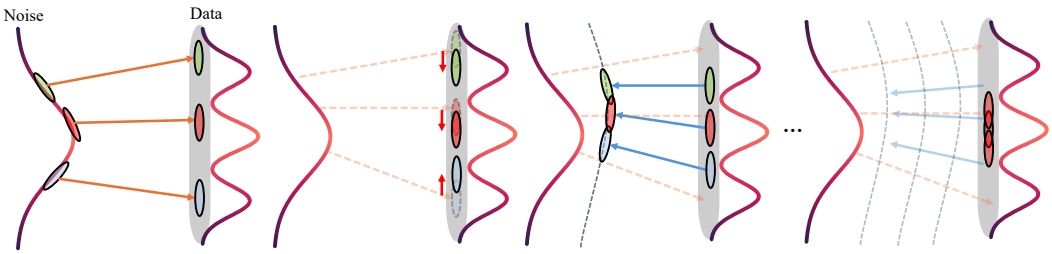
which can also be equivalently derived from Tweedie's formula (Efron, 2011). Then, an Euler step (4) can be reformulated as (Kim et al., 2025):

$$\boldsymbol{x}_s = (1-s)\hat{\boldsymbol{x}}_{0|t} + s\hat{\boldsymbol{x}}_{1|t}, \tag{7}$$

which corresponds to the interpolation between denoised and noisy estimates. For a text-guided flow model, the training objective is often given by:

$$\mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{x}_1}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t,t,\boldsymbol{c}) - \boldsymbol{v}(\boldsymbol{x}_t|\boldsymbol{x}_0)\|^2\right], \tag{8}$$

where $\boldsymbol{c}$ represents the textual embedding. Throughout this paper, we will often omit $\boldsymbol{c}$ from $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ or $\hat{\boldsymbol{x}}_{0|t}(\boldsymbol{c}) = \boldsymbol{x}_t - t\boldsymbol{v}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ if it does not lead to notational ambiguity. Following standard practice, we consider the latent flow model with a pretrained encoder-decoder $(\mathcal{E}_\phi, \mathcal{D}_\psi)$, and with a slight abuse of notation, continue to use $\boldsymbol{x}$ to denote encoded video latents throughout.



(a) Denoised estimation (b) Tweedie Matching (c) Stochastic Sampling (d)Final stage

Figure 2: **Main pipeline.** We generate long videos in inference-time by harmonizing multiple short video chunk sampling trajectories. **(a)** Each chunk starts from independent noise and follows its own ODE, producing divergent denoised estimates. **(b)** Tweedie matching corrects each chunk's denoised estimate toward its neighbor's on their shared frames (Sec. 4.1). **(c)** In early sampling phase, we inject stochastic noise to prevent trajectories from reverting to their original divergent ODE paths (Sec 4.2). **(d)** Repeating (b)–(c) progressively synchronizes all chunks into a coherent long video.

4 FLOWLONG: INFERENCE-TIME LONG VIDEO GENERATION

Given a pretrained model $\boldsymbol{v}_\theta$ over F-frame chunk latents $\boldsymbol{x} \in \mathbb{R}^{F\times d}$, our goal is to generate a coherent long video $\boldsymbol{X} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^K)$ of $N > F$ frames without fine-tuning. To this end, we simultaneously sample a chain of K chunks and harmonize them into a temporally consistent sequence. Each chunk $\boldsymbol{x}^k$ is conditioned on its own text prompt $\boldsymbol{c}_k$, which may vary across chunks. Note that $\boldsymbol{v}_\theta$ is invoked independently on each chunk at every sampling step, never on the full sequence.

Towards training-free long video generation, the key challenge is: *How to synchronize frame transitions across different chunk sampling trajectories that may diverge due to independent ODE noise initializations or potentially distinct prompts*? To answer this question, we adopt a fundamentally different strategy by formulating long video generation as an optimization problem. Specifically, our framework is based on neighbor-chunk conditioned latent optimization objective, which, when minimized during the reverse sampling process, progressively aligns each adjacent video chunks for smooth transitions. To prevent early divergence and facilitate mixing across trajectories, we further cast the initial sampling phase as an SDE by injecting stochastic noise, transitioning to deterministic ODE sampling in later stages. More details follows.

4.1 TWEEDIE MATCHING

For coherent long video sequence generation, we impose the following constraint – adjacent video chunk latents $\boldsymbol{x}^k$ and $\boldsymbol{x}^{k+1}$ should share a consistent overlap of O frames: within this overlap window, the last O frames of chunk k should coincide with the first O frames of chunk $k+1$. To formalize this constraint, let $\mathbf{1}_{\Omega_k}, \mathbf{1}_{\Omega'_{k+1}} \in \{0,1\}^F$ denote the indicator vectors:

$$\mathbf{1}_{\Omega_k} = (\underbrace{0,\ldots,0}_{F-O},\underbrace{1,\ldots,1}_{O})^\top, \qquad \mathbf{1}_{\Omega'_{k+1}} = (\underbrace{1,\ldots,1}_{O},\underbrace{0,\ldots,0}_{F-O})^\top, \tag{9}$$

which gives corresponding frame-selection matrices $M_k, M'_{k+1} \in \mathbb{R}^{O\times F}$ as follows:

$$M_k = \big[\, 0_{O\times(F-O)} \;\big|\; I_O \,\big], \qquad M'_{k+1} = \big[\, I_O \;\big|\; 0_{O\times(F-O)} \,\big]. \tag{10}$$

Both map a chunk $\boldsymbol{x}$ into the shared overlap window $\mathbb{R}^{O\times d}$, where $M_k^\top M_k = \mathrm{diag}(\mathbf{1}_{\Omega_k})$ and $M_{k+1}'^\top M'_{k+1} = \mathrm{diag}(\mathbf{1}_{\Omega'_{k+1}})$. Then, the hard overlap constraint reads:

$$M_k\, \boldsymbol{x}_0^{(k)} = M'_{k+1}\, \boldsymbol{x}_0^{(k+1)}, \qquad k = 1,\ldots,K-1. \tag{11}$$

Guidance loss. We relax (11) into a sampling guidance loss ℓ_k defined on the clean manifold. At time t and k-th chunk, ℓ_k is defined as:

$$\ell_k(\boldsymbol{x}; t) = \frac{1}{2}\big\| M_k\, \boldsymbol{x} - M'_{k+1}\, \hat{\boldsymbol{x}}_{0|t}^{(k+1)}(\boldsymbol{c}_{k+1}) \big\|^2, \tag{12}$$



Figure 3: **Qualitative comparison of 30-second video generation.** Baselines suffer from repetitive motion patterns and drift errors caused by accumulated exposure bias. In contrast, our method produces videos with diverse motion dynamics and is robust to exposure bias. Please refer to the supplementary video.

where $\hat{\boldsymbol{x}}_{0|t}^{k+1}(\boldsymbol{c}_{k+1})$ refers to the clean estimate of adjacent chunk $k+1$ as in (5), $\boldsymbol{x} \in \mathcal{M}$ with a clean data manifold $\mathcal{M}$. This guidance loss represents an ideal overlap condition that neighboring video chunk latents should satisfy. This formulation is structurally identical to the inverse problem template $\frac{1}{2}\|\boldsymbol{y} - \mathcal{A}\boldsymbol{x}\|^2$ in diffusion inverse solvers (Chung et al., 2023), with forward operator $\mathcal{A} = M_k$ and measurement $\boldsymbol{y} = M'_{k+1}\hat{\boldsymbol{x}}_{0|t}^{(k+1)}(\boldsymbol{c}_{k+1})$ given by the neighboring chunk.

Latent optimization. Following diffusion inverse solvers (DDS (Chung et al., 2023)), we can now integrate the optimization step of ℓ_k in terms of denoised estimates $\hat{\boldsymbol{x}}_{0|t}^{k}(\boldsymbol{c}_k)$, resulting in a modulated Euler step ($t \to s$):

$$\begin{aligned} \bar{\boldsymbol{x}}_{0|t}^{k}(\boldsymbol{c}_k) &:= \hat{\boldsymbol{x}}_{0|t}^{k}(\boldsymbol{c}_k) - \gamma_t \nabla_{\hat{\boldsymbol{x}}_{0|t}^{k}(\boldsymbol{c}_k)} \ell_k(\hat{\boldsymbol{x}}_{0|t}^{k}(\boldsymbol{c}_k); t) \\ \boldsymbol{x}_s^k &= (1-s)\bar{\boldsymbol{x}}_{0|t}^{k}(\boldsymbol{c}_k) + s\bar{\boldsymbol{x}}_{1|t}^{k}(\boldsymbol{c}_k), \end{aligned} \tag{13}$$

where $\bar{\boldsymbol{x}}_{1|t}^{k}(\boldsymbol{c}_k) = \frac{\boldsymbol{x}_t - (1-t)\bar{\boldsymbol{x}}_{0|t}(\boldsymbol{c}_k)}{t}$ as in (5). The gradient guidance is delineated as follows:

$$\nabla_{\hat{\boldsymbol{x}}_{0|t}^{(k)}} \ell_k(\hat{\boldsymbol{x}}_{0|t}^{k}(\boldsymbol{c}_k); t) = M_k^\top \big(M_k\, \hat{\boldsymbol{x}}_{0|t}^{(k)}(\boldsymbol{c}_k) - M'_{k+1}\, \hat{\boldsymbol{x}}_{0|t}^{(k+1)}(\boldsymbol{c}_{k+1})\big), \tag{14}$$

which is supported only on the overlap frames. Specifically, (13) is reformulated as:

$$\bar{\boldsymbol{x}}_{0|t}^{(k)} = \hat{\boldsymbol{x}}_{0|t}^{(k)}(\boldsymbol{c}_k) - \lambda\, M_k^\top \big(M_k\, \hat{\boldsymbol{x}}_{0|t}^{(k)}(\boldsymbol{c}_k) - M'_{k+1}\, \hat{\boldsymbol{x}}_{0|t}^{(k+1)}(\boldsymbol{c}_{k+1})\big), \tag{15}$$

where $\lambda > 0$ absorbs the step size γ_t. Per frame, since $M_k^\top M_k = \mathrm{diag}(\mathbf{1}_{\Omega_k})$, this update reads

$$\bar{\boldsymbol{x}}_{0|t}^{(k)}[j] = \begin{cases} \hat{\boldsymbol{x}}_{0|t}^{(k)}(\boldsymbol{c}_k)[j], & j \notin \Omega_k, \\ (1-\lambda_j)\, \hat{\boldsymbol{x}}_{0|t}^{(k)}(\boldsymbol{c}_k)[j] + \lambda_j\, \hat{\boldsymbol{x}}_{0|t}^{(k+1)}(\boldsymbol{c}_{k+1})[j'], & j \in \Omega_k, \end{cases} \tag{16}$$

where $j' = j - (F - O)$ is the corresponding frame index in Ω'_{k+1}, and λ_j refers to per-frame step size. Non-overlap frames ($j \notin \Omega_k$) remain untouched, while overlap frames are interpolated toward each neighbor's denoised estimate from Tweedie's formula. We call this update *Tweedie matching*, which is manifold-constrained due to the use of DDS (Chung et al., 2023). A symmetric update is applied to chunk $k+1$ and others. In practice, we set λ_j to a symmetric schedule over the overlap window, ensuring smooth frame-level matching and exact consistency at the boundary, so that each overlap region is stored once and shared by both chunks without duplication. See the appendix for details.

Inverse-problem as a design principle. To our best, FlowLong is the first to formulate long video generation as an inverse problem, which allows us to adopt design principles of diffusion inverse solvers (Daras et al., 2024; Chung et al., 2022; 2023). A key principle is the optimization space: for instance, Chung et al. (2023) relates the update of ℓ_k on the denoised estimate $\hat{\boldsymbol{x}}_{0|t}(\boldsymbol{c}_k)$ to a projected gradient on the clean data manifold, supporting our manifold-constrained update. In line with this connection, we show that alternative optimization spaces (e.g., noisy latents $\boldsymbol{x}_t$ as in MultiDiffusion (Bar-Tal et al., 2023)) underperform in Table 3, implying that the choice of optimization space is critical to synchronization quality. We believe the formulation further opens a broader design space (e.g., additional regularizers, latent rewards, perceptual losses), which we leave for future work.

Prompt conditioning. When all chunks share a common prompt ($\boldsymbol{c}_k = \boldsymbol{c}$ for all k), the guidance loss (12) enforces temporal coherence under a single scene description. For multi-shot generation with per-chunk prompts $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$, we condition each chunk on a shared global prompt $\boldsymbol{c}_{\text{global}}$ to maintain stylistic and semantic consistency across scene transitions, while the additional per-chunk prompt supplements local content.

4.2 Stochastic Early-Phase Sampling

While Sec. 4.1 regularizes the denoising paths toward a coherent long video sequence, under a deterministic ODE sampling regime, this correction may be insufficient to fully synchronize video chunks. Specifically, even after the clean estimate is pulled toward the neighbor via Tweedie matching, the deterministic renoising step drives $\boldsymbol{x}_s^k$ back toward the original ODE trajectory. When ODE trajectories are initialized from independent Gaussian noise (and conditioned on potentially distinct prompts) their trajectories may be far apart in latent space, and this inertia prevents the long video harmonization across time steps.

To break this inertia, we inject stochastic noise during the early sampling phase by casting the renoising step in stochastic form. The injected noise perturbs each chunk away from its deterministic trajectory, effectively renoising the state after each Tweedie matching correction. Following FlowDPS (Kim et al., 2025), we mix the stochastic noise in (13) as:

$$\boldsymbol{x}_s^k = (1-s)\bar{\boldsymbol{x}}_{0|t}^k(\boldsymbol{c}_k) + s\tilde{\boldsymbol{x}}_{1|t}^k(\boldsymbol{c}_k), \tag{17}$$

where

$$\tilde{\boldsymbol{x}}_{1|t}^k(\boldsymbol{c}_k) = \sqrt{1-\eta_t}\bar{\boldsymbol{x}}_{1|t}^k(\boldsymbol{c}_k) + \sqrt{\eta_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{18}$$

By setting $\kappa_{s,t} = s\sqrt{\eta_t}$, the renoising step in (17) can be reformulated in stochastic form as follows:

$$\boldsymbol{x}_s^k = (1-s)\bar{\boldsymbol{x}}_{0|t}^k(\boldsymbol{c}_k) + \sqrt{s^2-\kappa_{s,t}^2}\bar{\boldsymbol{x}}_{1|t}^k(\boldsymbol{c}_k) + \sqrt{\kappa_{s,t}^2}\boldsymbol{\epsilon}, \tag{19}$$

which decomposes the renoising into a deterministic component along $\bar{\boldsymbol{x}}_{1|t}^k(\boldsymbol{c}_k)$ and a stochastic perturbation of magnitude $\kappa_{s,t}$.

In practice, we adopt a binary schedule $\eta_t = \mathbb{1}(t \geq t^*)$ for a threhold t^*. This implies that the early stochastic phase ($t \geq t^*$) uses full stochastic renoising to remix trajectories after each Tweedie matching correction, while the later phase ($t < t^*$) reverts to deterministic ODE sampling to preserve fine-grained visual fidelity. As shown in Figure 3, experimental results demonstrate that this hybrid sampling approach significantly improves temporal consistency and mitigates exposure bias in long video generation. Exploring smoother schedules for η_t is an interesting direction for future work.

4.3 Extend to other generation tasks

Our framework is not specific to temporal extension of visual video models; it applies broadly to any setting where a pretrained flow model generates fixed-size windows and the goal is to produce outputs that exceed this native horizon. The key requirement is that adjacent windows share an overlap region where Tweedie matching can enforce consistency. As promising examples, we demonstrate two additional applications: audio-video joint generation and text-to-3D generation. Crucially, *none* of

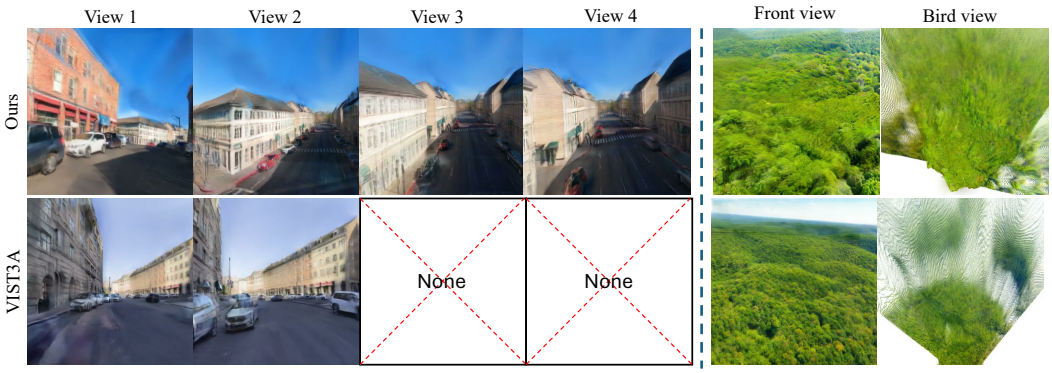


Figure 4: **Qualitative comparison of text-to-3DGS generation.** VIST3A (Go et al., 2026) is confined to the native window of its video model and cannot render views beyond it, whereas our method covers substantially wider viewpoints from the same text.

these extensions require *fine-tuning*, in contrast to existing autoregressive long video models that must be retrained for each backbone and task.

Audio-video joint generation. LTX-2 (HaCohen et al., 2026) is a flow-matching video DiT augmented with an audio branch and cross-modal attention, denoising video and audio latents jointly under a shared text condition. To extend it beyond its native window, we decompose each modality into a chain of K chunks aligned in time through the model's frame-rate ratio, and apply Tweedie matching (Sec. 4.1) to both streams with the same matching schedule λ_j. The corrected estimates are then advanced by stochastic early-phase renoising (Sec. 4.2) with independent perturbations ϵ^v, ϵ^a per modality, producing arbitrarily long, phase-locked audio-video sequences without any fine-tuning.

Text-to-3D generation. VIST3A (Go et al., 2026) stitches a feed-forward 3D reconstructor, AnySplat (Jiang et al., 2025), into the latent space of Wan 2.1 (Wan et al., 2025) via a lightweight bridge layer, converting a denoised video latent into 3D Gaussian splats in a single forward pass without per-scene optimization.We sample a chain of K independent noise chunks spanning the target length, and apply Tweedie matching (Sec. 4.1) followed by stochastic early-phase renoising (Sec. 4.2) at every sampling step. The resulting extended video latent is then decoded and fed to AnySplat, producing a longer 3D scene from text alone.

5 Experiments

For long video generation, we compare against bidirectional diffusion models (RIFLEx (Zhao et al., 2025a), UltraViCo (Zhao et al., 2025b)) and autoregressive diffusion models (CausVid (Yin et al., 2025), Self-Forcing (Huang et al., 2025), Deep-Forcing (Yi et al., 2025), ∞ -RoPE (Yesiltepe et al., 2025), LongLive (Yang et al., 2025)), and against VIST3A (Go et al., 2026) for text-to-3DGS generation. We evaluate using VBench (Huang et al., 2024) across seven dimensions: aesthetic quality, imaging quality, background consistency, subject consistency, motion smoothness, dynamic degree, and temporal flickering, generating 30s and 60s videos from 100 MovieGen Bench (Polyak et al., 2024) prompts and 100 SceneBench (Yuanbo et al., 2024) prompts for 3DGS. Our method is applied without additional training on Wan 2.1-T2V-1.3B (Wan et al., 2025) and LTX-2 (HaCohen et al., 2026) for long video generation, and Wan 2.1-T2V-14B with AnySplat (Jiang et al., 2025) for text-to-3DGS, all on a single NVIDIA H100 GPU.

5.1 Long video generation

Qualitative results. We provide a qualitative comparison of 30s video generation in Figure 3. For bidirectional models (Zhao et al., 2025b;a), meaningful motion nearly vanishes and pixel values become saturated at 30 seconds. A similar phenomenon is observed in autoregressive models (Yin et al., 2025; Huang et al., 2025; Yesiltepe et al., 2025; Yang et al., 2025; Yi et al., 2025), where pixel values progressively saturate over time, leading to error drift. Furthermore, since these models

Table 1: **Performance comparisons on 30s and 60s videos.** The first and second values are highlighted. We report VBench scores for each baseline, categorized by model type.

Model	Results on 30s ↑								Results on 60s ↑							
	Aesthetic Quality	Background Consistency	Dynamic Degree	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Overall↑	Aesthetic Quality	Background Consistency	Dynamic Degree	Imaging Quality	Motion Smoothness	Subject Consistency	Temporal Flickering	Overall↑
Bidirectional models -1.3B																
RIFLEx	0.44	**0.97**	0.08	0.41	**0.99**	**0.97**	**0.99**	0.6943	-	-	-	-	-	-	-	-
UltraViCo	0.4708	0.9348	0.5612	0.4334	0.9895	0.8793	0.9866	0.7508	-	-	-	-	-	-	-	-
Wan2.1 + Ours	**0.5777**	0.9305	**0.7800**	**0.6368**	0.9877	0.8751	0.9753	**0.8233**	-	-	-	-	-	-	-	-
Bidirectional models - 14B																
Ltx2	**0.5412**	0.8845	**0.6251**	0.6124	0.9813	0.8152	0.9482	0.7733	-	-	-	-	-	-	-	-
Ltx2 + Ours	0.5337	**0.9016**	0.6162	**0.6393**	**0.9852**	**0.8201**	**0.9773**	**0.7812**	-	-	-	-	-	-	-	-
Autoregressive models-1.3B																
CausVid	0.5773	0.9037	0.4545	0.6556	0.9819	0.8874	0.9718	0.7760	0.5746	0.8871	0.4242	0.6438	0.9815	0.8613	0.9723	0.7636
Self-Forcing	0.5523	0.9064	0.5455	0.6893	0.9858	0.8760	0.9752	0.7901	0.5355	0.8690	0.4747	0.6632	0.9854	0.8056	0.9776	0.7587
Deep-Forcing	0.5667	0.9280	0.6566	0.6872	0.9836	0.9019	0.9718	0.8137	0.5691	0.9310	0.5253	0.6831	0.9850	0.9099	0.9745	0.7968
∞-RoPE	0.5724	0.9352	0.5102	0.6763	0.9870	0.9128	0.9765	0.7958	0.5640	0.9294	0.5102	0.6803	0.9868	0.9046	0.9764	0.7931
LongLive	**0.5868**	**0.9453**	0.3535	**0.6967**	**0.9895**	**0.9294**	**0.9792**	0.7829	**0.5885**	**0.9413**	0.4141	**0.6916**	**0.9891**	**0.9279**	**0.9784**	0.7902
Wan2.1 + Ours	0.5777	0.9305	**0.7800**	0.6368	0.9877	0.8751	0.9753	**0.8233**	0.5738	0.9213	**0.8200**	0.6391	0.9869	0.8605	0.9738	**0.8251**

Table 2: **Long-range drift on 60s videos.** We measure drift with complementary metrics; benchmark details are in Appendix B.1.

Metric	FlowLong	CausVid	Self-Forcing	Deep-Forcing	∞-RoPE	LongLive
Visual Stability ↑	97.2	74.3	46.7	97.6	99.0	**99.2**
Semantic Drift ↓	**0.0003**	0.0033	0.0093	0.0010	0.0031	0.0009
Content Retention ↑	0.927	0.934	0.817	**0.938**	0.933	0.927
Dynamic Degree ↑	**0.8200**	0.4242	0.4747	0.5253	0.5102	0.4141

continuously cache the key-value pairs of previous frames, the diversity of motion is severely limited, resulting in repetitive motion patterns. In contrast, our method regularizes and samples videos from independent initial points, which enables rich motion diversity and effectively eliminates the error drift that accumulates over time. Additionally, as shown in Figure 6 (Appendix B.4), our method supports multi-shot generation by combining a global prompt with per-chunk local prompts, enabling diverse and semantically coherent scene transitions across extended video sequences.

Quantitative results on VBench. Table 1 reports VBench scores for 30s and 60s video generation, organized into three groups by model type. Against training-free bidirectional models (RIFLEx, UltraViCo), our method achieves the best overall score, with superior results including Dynamic Degree. For LTX2, where no training-free method supports generation beyond 30 seconds, we compare against a sliding-window baseline and achieve higher overall scores across most metrics. In comparison with autoregressive models, our model outperforms all baselines by a substantial margin on Dynamic Degree, yielding the best overall performance and demonstrating robust motion diversity in long video generation. Some AR baselines lead on consistency metrics, but by small margins: LongLive at 60s leads by +0.0022 motion smoothness and +0.0046 temporal flickering while generating half the motion (0.41 vs. 0.82). VBench itself notes that near-static videos can "cheat" its temporal consistency metrics (Huang et al., 2024), and Self-Forcing++ (Cui et al., 2025) attributes the high temporal scores of AR baselines to motion stagnation. In contrast, FlowLong preserves the rich motion of Wan 2.1, whose Dynamic Degree is 0.89 on its native 5s clips: ours retains 0.82 at 60s because no window is conditioned on self-generated history, whereas the AR baselines distilled from the same backbone collapse to 0.41–0.53 under KV-cache feedback. The added motion is coherent, as motion smoothness (0.9869) and temporal flickering (0.9738) stay on par with all baselines and Table 2 shows no long-range drift.

Long-range drift. Table 2 quantifies drift on 60s videos with three complementary metrics. Early AR models drift severely (Visual Stability: CausVid 74.3, Self-Forcing 46.7), whereas FlowLong reaches 97.2, on par with recent AR methods explicitly engineered against drift (97.6–99.2). FlowLong also records the lowest Semantic Drift (0.0003) and a Content Retention within the band of the strongest baselines (0.927 vs. 0.927–0.938), while producing roughly twice their motion. A user study (Appendix B.2) agrees: FlowLong is preferred in 84–92% of pairwise comparisons on semantic consistency, motion diversity, and color saturation.

5.2 TEXT-TO-3DGS

Fig. 4 compares VIST3A with our extended pipeline. Bound to the fixed window of its video model, the baseline yields few Gaussians with limited coverage, while ours reconstructs longer scenes with wider viewpoints and denser geometry. Quantitatively (Fig. 10 in Appendix D; 100 SceneBench prompts, AnySplat depth confidence), our method produces 1.64× more Gaussians per scene, keeps 2.47M after top-30% confidence filtering, and raises the mean confidence logit from 26.27 to 41.52.



Figure 5: **Qualitative ablation.** Comparison of denoised estimates and final generations across sampling strategies. The spatial layout is largely determined in the early sampling stages (top row). Full SDE ($\eta_t = 1$ in (18)) preserves temporal consistency across chunks but degrades visual quality, while full ODE ($\eta_t = 0$) yields sharper results at the cost of temporal inconsistency.

5.3 Ablation Study

We ablate the two key components of FlowLong in Table 3.

Optimization space. Matching overlaps on the denoised estimates $\hat{\boldsymbol{x}}_{0|t}$ outperforms matching noisy latents $\boldsymbol{x}_t$ (as in MultiDiffusion (Bar-Tal et al., 2023)), on consistency and quality with comparable motion (More discussions in Appendix C).

Table 3: **Ablation study.** The first and second values are highlighted. x_t matching differs from ours only in the fusion space.

Method	Consistency ↑	Motion ↑	Quality ↑
Full SDE sampling	0.9427	0.9449	0.5298
Full ODE sampling	0.9604	0.9621	0.6075
x_t matching	0.9579	**0.9690**	0.5862
Ours	**0.9615**	0.9685	**0.6359**

Sampling strategy. Figure 5 compares three noise schedules η_t in (18): full SDE ($\eta_t = 1$), full ODE ($\eta_t = 0$), and ours ($\eta_t = \mathbb{1}(t \geq t^*)$). The denoised estimates (top row of Fig. 5) reveal that the global layout of each chunk is settled in the early sampling steps, so chunks must be synchronized before this point. Under full ODE, each chunk drifts back to its own trajectory after every correction (Sec. 4.2), leaving adjacent chunks sharp but mismatched in content. Full SDE keeps chunks aligned, but noise injected in the late steps continually disturbs visual details, washing out texture and lowering all metrics. Our schedule injects noise only in early phase, achieving both cross-chunk alignment and fine detail with the best consistency.

5.4 Computational Costs

Our framework reduces to a simple closed-form interpolation between neighboring chunks, introducing negligible overhead. This contrasts with bidirectional alternatives such as RIFLEx (Zhao et al., 2025a) and UltraViCo (Zhao et al., 2025b), which rely on architecture-specific modifications. Moreover, being backbone-agnostic, FlowLong readily composes with distilled few-step backbones (e.g., 4-step), reducing the total cost to a level comparable to autoregressive alternatives: on 30s generation, FlowLong takes 168 s at 21 GB with a 4-step backbone versus 77–113 s for AR baselines, and 13 min versus 21–25 min for UltraViCo and RIFLEx at less than half their memory (Appendix B.3).

6 Conclusion

We presented FlowLong, a training-free, architecture-agnostic framework for long video generation built on two core components: Tweedie matching, which enforces temporal consistency across adjacent chunks by matching predicted clean samples, and stochastic early-phase sampling, which synchronizes per-window trajectories by injecting stochastic noise in the high-noise regime before transitioning to deterministic ODE sampling. These components address the key failure modes of prior approaches without any architectural modifications or additional training. We validated FlowLong across text-to-video, audio-video joint generation, and text-to-3DGS tasks, consistently outperforming both training-free and autoregressive baselines. One limitation is that our consistency constraint between neighboring chunks is inherently local, which may hinder global semantic coherence in extremely long videos, and we leave this as future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing and debugging the code. Other tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript for readability and to assist in creating figures. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work proposes an inference-time sampling procedure for pretrained video generative models and does not involve human subjects, private data, or new data collection. All experiments use publicly released models (Wan 2.1, LTX-2, VIST3A/AnySplat) and public prompt benchmarks (MovieGen Bench, SceneBench). As with any method that extends the capability of video generation models, longer and more coherent synthetic videos could be misused to produce misleading content; our method inherits the content safeguards and usage terms of the underlying pretrained models and does not remove or weaken them. We do not foresee additional ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

The proposed method is training-free and fully specified by the equations in Sec. 4.1 and Sec. 4.2. Appendix A provides the latent-space window geometry, the explicit form of the matching schedule λ_j, the multi-chunk aggregation algorithm used in our implementation, and the corresponding geometry for audio–video joint generation. Sec. 5 lists the backbones, benchmarks, evaluation metrics, video lengths, and hardware used for all reported numbers. Source code and the exact sampling configurations will be released to facilitate reproduction.

REFERENCES

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14834–14844, 2025.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.

Jake Bruce, Michael D 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 *Forty-first International Conference on Machine Learning*, 2024.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in neural information processing systems*, 35:25683–25696, 2022.

Hyungjin Chung, Suhyeon Lee, and Jong Chul Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. *arXiv preprint arXiv:2303.05754*, 2023.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025.

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Hyojun Go, Dominik Narnhofer, Goutam Bhat, Prune Truong, Federico Tombari, and Konrad Schindler. Text-to-3d by stitching a multi-view reconstruction network to a video generator. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kI27Niy4xY.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Yeobin Hong, Suhyeon Lee, Hyungjin Chung, and Jong Chul Ye. Inversecrafter: Efficient video recapture as a latent domain inverse problem. *arXiv preprint arXiv:2512.05672*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Hyeonho Jeong, Suhyeon Lee, and Jong Chul Ye. Reangle-a-video: 4d video generation as video-to-video translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11164–11175, 2025.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

Jaihoon Kim, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. Synctweedies: A general generative framework based on synchronized diffusions. *arXiv:2403.14370*, 2024a.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Flowdps: Flow-driven posterior sampling for inverse problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12328–12337, 2025.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. In *NeurIPS*, 2024b.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Jangho Park, Taesung Kwon, and Jong Chul Ye. Zero4d: Training-free 4d video generation from single video using off-the-shelf video diffusion model. *arXiv e-prints*, pp. arXiv–2503, 2025.

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

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Junyoung Seo, Hyunwook Choi, Minkyung Kwon, Jinhyeok Choi, Siyoon Jin, Gayoung Lee, Junho Kim, JoungBin Lee, Geonmo Gu, Dongyoon Han, et al. Grounding world simulation models in a real-world metropolis. *arXiv preprint arXiv:2603.15583*, 2026.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitrii Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. SV3D: Novel multi-view synthesis and 3D generation from a single image using latent video diffusion. In *European Conference on Computer Vision (ECCV)*, 2024.

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.

Chaoyang Wang, Peiye Zhuang, Tuan Duc Ngo, Willi Menapace, Aliaksandr Siarohin, Michael Vasilkovsky, Ivan Skorokhodov, Sergey Tulyakov, Peter Wonka, and Hsin-Ying Lee. 4real-video: Learning generalizable photo-realistic 4d video diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17723–17732, 2025.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26057–26068, 2025.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. *arXiv preprint arXiv:2511.20649*, 2025.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *CVPR*, 2025.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 100–111, 2025.

Yang Yuanbo, Shao Jiahao, Li Xinyang, Shen Yujun, Geiger Andreas, and Liao Yiyi. Prometheus: 3d-aware latent diffusion models for feed-forward text-to-3d scene generation. *arxiv:2412.21117*, 2024.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame context packing and drift prevention in next-frame-prediction video diffusion models. *arXiv preprint arXiv:2504.12626*, 2025a.

Lvmin Zhang, Shengqu Cai, Muyang Li, Chong Zeng, Beijia Lu, Anyi Rao, Song Han, Gordon Wetzstein, and Maneesh Agrawala. Pretraining frame preservation in autoregressive video memory compression. *arXiv e-prints*, pp. arXiv–2512, 2025b.

Min Zhao, Guande He, Yixiao Chen, Hongzhou Zhu, Chongxuan Li, and Jun Zhu. Riflex: A free lunch for length extrapolation in video diffusion transformers. *arXiv preprint arXiv:2502.15894*, 2025a.

Min Zhao, Hongzhou Zhu, Yingze Wang, Bokai Yan, Jintao Zhang, Guande He, Ling Yang, Chongxuan Li, and Jun Zhu. Ultravico: Breaking extrapolation limits in video diffusion transformers. *arXiv preprint arXiv:2511.20123*, 2025b.

A TWEEDIE MATCHING: IMPLEMENTATION DETAILS

In this section, we expand the practical details of Sec. 4.1. We describe the latent-space window geometry (§A.1), the explicit form of the per-frame matching schedule λ_j (§A.2), an equivalence between iterating Eq. (15) over all chunk pairs and a single weighted aggregation (§A.3), the aggregation algorithm used in our implementation (§A.4), and the corresponding geometry for audio–video joint generation (§A.5).

A.1 WINDOW GEOMETRY IN LATENT SPACE

All operations are performed in the latent space of a video VAE with temporal stride r ($r = 8$ for LTX-2). We denote

- F: number of latent frames per chunk (window length);
- O: number of latent frames in the *matching zone*—the last O frames of each chunk, on which Tweedie matching is applied;
- S: stride between the global start indices of consecutive chunks;
- K: number of chunks (consistent with Sec. 4.1);
- $N = F + (K-1)\,S$: total number of latent frames in the long-video buffer.

We require

$$O \;\geq\; S, \tag{20}$$

so that every latent frame in chunk k's matching zone is also predicted by chunk $k+1$ at the same global temporal position; otherwise $M'_{k+1}\hat{\boldsymbol{x}}_{0|t}^{(k+1)}$ in the guidance loss (12) would refer to frames that chunk $k+1$ never observed.

In practice the user specifies the pixel-space window size W and the pixel index w at which the overlap region begins inside a window. The latent quantities are then obtained by

$$F = \tfrac{W-1}{r} + 1, \qquad S = \big\lfloor (W-w)/r \big\rfloor, \qquad O = F - \lfloor w/r \rfloor. \tag{21}$$

For the LTX-2 backbone with $W = 121$ and $w = 64$, this yields $(F, O, S) = (16, 8, 7)$, satisfying (20) with one frame of slack; we use this configuration throughout our text-to-video experiments.

We index the global latent buffer as $g \in \{0, \ldots, N-1\}$ and the local frames within chunk $k \in \{1, \ldots, K\}$ as $j \in \{0, \ldots, F-1\}$, related by $g = (k-1)\,S + j$. Chunk k's matching zone in local indexing is therefore

$$\Omega_k \;=\; \{\, F-O,\; F-O+1,\; \ldots,\; F-1 \,\}, \tag{22}$$

in agreement with the indicator vector $\mathbf{1}_{\Omega_k}$ in Sec. 4.1.

A.2 THE PER-FRAME MATCHING SCHEDULE λ_j

On the matching zone, Eq. (15) reduces to a per-frame convex combination between the two adjacent clean estimates with weight λ_j. We adopt the linear schedule

$$\lambda_j \;=\; \frac{j-(F-O)}{O-1}, \qquad j \in \Omega_k, \tag{23}$$

which has three properties used throughout the rest of the appendix.

1. **Boundary consistency.** $\lambda_{F-O} = 0$ and $\lambda_{F-1} = 1$. Hence the leftmost frame of the matching zone is taken entirely from chunk k, and the rightmost frame entirely from chunk $k+1$. The matched estimate $\bar{\boldsymbol{x}}_{0|t}^{(k)}$ therefore agrees exactly with $\hat{\boldsymbol{x}}_{0|t}^{(k)}$ at the seam $j = F-O$ and exactly with $\hat{\boldsymbol{x}}_{0|t}^{(k+1)}$ at $j = F-1$, eliminating any discontinuity at either side of the overlap.
2. **Symmetry.** Setting the local index $i = j-(F-O) \in \{0, \ldots, O-1\}$, we have $\lambda_j = i/(O-1)$ and the mirror identity

$$\lambda_j + \lambda_{j_{\mathrm{mir}}} \;=\; 1, \qquad j_{\mathrm{mir}} \;=\; (F-O) + (O-1) - i \;=\; F-1-i.$$

Equivalently, chunk k's update applies weight λ_j to chunk $k+1$'s prediction, while the symmetric update applied to chunk $k+1$ (Sec. 4.1) applies weight $1-\lambda_j$ to chunk k's prediction at the same global frame. Both updates therefore produce the *same* convex combination $(1-\lambda_j)\,\hat{\boldsymbol{x}}_{0|t}^{(k)}[j] + \lambda_j\,\hat{\boldsymbol{x}}_{0|t}^{(k+1)}[j']$. We exploit this in §A.3 to compute the aggregation only once.

3. **Smoothness.** λ_j is linear in the frame index, so frame-level transitions across the matching zone are uniform. We did not observe additional gains from smoother schedules (e.g. raised-cosine windows) in preliminary experiments, while the linear form admits the simple equivalence in §A.3.

A.3 Multi-chunk pairwise updates collapse to a single weighted aggregation

Sec. 4.1 states the modulated Euler step (13) for an adjacent pair $(k, k+1)$. With K chunks one would, in principle, iterate this update over all pairs $(1,2),(2,3),\ldots,(K-1,K)$ and average where matching zones overlap. Under the symmetric linear schedule (23), this iteration collapses into a single pass that produces, for each global frame g, a weighted average of *exactly one* adjacent pair of clean predictions.

Partition the global indices $\{0,\ldots,N-1\}$ as

- the *leading prefix* $g \in [0,\ F-O)$, owned solely by chunk 1;
- per-pair *matching zones* $\mathcal{B}_k \;=\; \big[\,(k-1)S+(F-O),\ \ (k-1)S+F\,\big)$ for $k=1,\ldots,K-1$;
- the *trailing suffix* $g \in [(K-1)S+(F-O),\ N)$, owned solely by chunk K;
- (only when $S>O$) an *interior gap* $\mathcal{G}_k = [(k-1)S+F,\ kS+(F-O))$ between $\mathcal{B}_k$ and $\mathcal{B}_{k+1}$, owned solely by chunk $k+1$.

For $g \in \mathcal{B}_k$, both chunks k and $k+1$ have a clean prediction at g, located at local indices $j = g-(k-1)S$ in chunk k and $j' = j-S$ in chunk $k+1$; the linear schedule (23) assigns weight $1-\lambda_j$ to chunk k and λ_j to chunk $k+1$. By Property 2 of §A.2, the symmetric update for chunk $k+1$ produces the same convex combination, so the two pairwise updates can be replaced by a single write of the convex combination into the global buffer.

When $S=O$ the matching zones $\{\mathcal{B}_k\}_{k=1}^{K-1}$ tile the global buffer cleanly. When $S<O$ (the typical case in our experiments, e.g. $(F,O,S)=(16,8,7)$), $\mathcal{B}_k$ and $\mathcal{B}_{k+1}$ overlap by $O-S$ frames; we resolve the conflict by using the *rightmost* pair, i.e. frames in $\mathcal{B}_k \cap \mathcal{B}_{k+1}$ are blended between chunks $k+1$ and $k+2$ (last-writer-wins; §A.4). This preserves the boundary-consistency property of §A.2: at every seam, the long-video latent is identical to one of the two adjacent chunk predictions, so each overlap frame is stored exactly once and shared between adjacent chunks, as stated in Sec. 4.1.

A.4 Aggregation in the long-video buffer

The reduction in §A.3 is realised by a single pass over the chunks. We allocate the long-video buffer $\hat{\boldsymbol{X}}_{0|t} \in \mathbb{R}^{N\times d}$ and write each global index exactly once: (i) the leading prefix $g \in [0,\ F-O)$ is copied from chunk 1; (ii) for each $k=1,\ldots,K-1$, the per-pair matching zone $\mathcal{B}_k$ is filled with $(1-\lambda_j)\hat{\boldsymbol{x}}_{0|t}^{(k)}[j] + \lambda_j \hat{\boldsymbol{x}}_{0|t}^{(k+1)}[j-S]$ for $j \in \Omega_k$, and any interior gap $\mathcal{G}_k$ that arises when $S>O$ is copied from chunk $k+1$; (iii) the trailing suffix $g \in [(K-1)S+(F-O),\ N)$ is copied from chunk K. Because the writes use direct assignment, no weight accumulation or re-normalisation is required, and overlaps between adjacent matching zones (the $S<O$ case in §A.3) are resolved by last-writer-wins. After the aggregation, the next-state update of Sec. 4.2 (deterministic Euler in the low-noise phase, stochastic renoising in the high-noise phase) is applied directly on the global buffer $\hat{\boldsymbol{X}}_{0|t}$, and the resulting $\boldsymbol{X}_s$ is then re-sliced into K overlapping windows for the next denoising step.

A.5 Audio–video joint geometry

LTX-2 (HaCohen et al., 2026) jointly denoises a video latent and an audio latent under a shared text condition (Sec. 4.1 and Sec. 4.2). The two streams have different temporal rates: the video latent rate is fps_v/r frames per second, while the audio latent rate is ρ_a latents per second ($\rho_a = 25$ for

LTX-2). To keep Tweedie matching temporally consistent across both modalities we choose the audio geometry that matches the video stride in seconds.

Given the video pixel window size W, the pixel overlap-start index w, and the video frame rate fps_v, the audio chunk length, stride, and overlap (all in audio latents) are

$$F_a = \mathrm{round}\left(\frac{W}{\mathrm{fps}_v}\,\rho_a\right),$$
$$S_a = \mathrm{round}\left(\frac{W-w}{\mathrm{fps}_v}\,\rho_a\right), \qquad O_a = F_a - S_a, \tag{24}$$

and the total audio length is $N_a = F_a + (K-1)\,S_a$. With $(W, w, \mathrm{fps}_v, \rho_a) = (121, 64, 24, 25)$ this gives $(F_a, O_a, S_a) = (126, 67, 59)$, which satisfies the same constraint $O_a \geq S_a$ as the video geometry (20).

The continuous-time rounding in (24) can introduce a misalignment of at most one audio latent ($\approx 1/\rho_a$ s, i.e. $40\,\mathrm{ms}$ for LTX-2) between the audio matching zone and the geometric overlap with the next chunk; we clamp this offset to zero in the aggregation, which we found to have no observable effect on audio–video phase locking. With this geometry, the aggregation of §A.4 is run independently on the video and audio buffers using their respective (F, O, S), and stochastic early-phase sampling (Sec. 4.2) is then applied with separate noise samples $\boldsymbol{\epsilon}^v$ and $\boldsymbol{\epsilon}^a$, so per-modality trajectories synchronise across windows while remaining temporally aligned with each other through the shared chunk geometry.

B Additional Result

B.1 Long-Range Drift Metrics

All three metrics in Table 2 are computed on the 60s videos of Table 1. *Visual Stability* follows the protocol of Self-Forcing++ (Cui et al., 2025): a video MLLM is prompted with a rubric that targets error accumulation and over-exposure and rates each video; ratings are aggregated to a 0–100 scale, where higher means less drift. *Semantic Drift* is the absolute difference between the CLIP prompt–frame similarity of the opening frames and that of the closing frames, so a low value means the video stays faithful to the prompt until the end. *Content Retention* is the CLIP similarity between the opening and closing frames themselves, so a high value means the initial subject and scene are preserved. The two protocols (MLLM rating and CLIP embeddings) identify the same degradation cases (CausVid, Self-Forcing), corroborating both measures. We view explicit long-term memory beyond the chunk chain as an open direction shared with minute-scale AR models (Cui et al., 2025).

B.2 User Study

To assess perceptual quality beyond automatic metrics, we ran a two-alternative forced choice (2AFC) study. Each participant compared FlowLong against five AR baselines (CausVid, Self-Forcing, Deep-Forcing, ∞-RoPE, LongLive) and, for each pair, chose the better clip along three axes: semantic consistency, motion diversity, and RGB saturation, with method identities hidden and presentation order randomized. Table 4 reports the win rate, i.e., the percentage of comparisons in which a method was preferred. FlowLong achieves the highest win rate on every axis, and no baseline exceeds 31% against it.

Table 4: **User study (2AFC) win rate against FlowLong across five AR baselines.**

Axis	FlowLong	CausVid	Self-Forcing	Deep-Forcing	∞-RoPE	LongLive
Semantic consistency ↑	**84.3%**	2.4%	9.5%	28.6%	7.1%	31.0%
Motion diversity ↑	**92.4%**	0.0%	4.8%	14.3%	2.4%	16.7%
RGB saturation ↑	**89.0%**	0.0%	4.8%	11.9%	7.1%	31.0%

B.3 Latency and Memory

Table 5 reports sampling time and peak VRAM for 30s generation on a single H100. Among training-free bidirectional methods on the standard 30-step backbone, FlowLong is the fastest and uses less than half the memory of UltraViCo and RIFLEx, because the model is invoked only on fixed-size windows and the footprint does not grow with the target length, whereas the baselines extend attention over the full sequence. The latency gap to AR baselines comes from the backbone: they are few-step generators obtained by distillation, whereas FlowLong runs a standard 30-step model without training. Applied to a 4-step distilled backbone (FlowLong*), FlowLong reaches the same order of latency as the AR baselines with unchanged memory and slightly higher quality, and it requires no re-distillation when a new backbone is released.

Table 5: **Latency and VRAM on 30s long video generation.** FlowLong* uses a 4-step distilled backbone.

	FlowLong	FlowLong*	UltraViCo	RIFLEx	CausVid	Self-Forcing	Deep-Forcing	∞-RoPE	LongLive
Sampling steps	30	4	30	30	3	4	4	4	4
Sampling time	13 m	168 s	21 m	25 m	77 s	113 s	103 s	79 s	101 s
VRAM	21 GB	21 GB	45 GB	51 GB	26.4 GB	20.6 GB	21.9 GB	16.8 GB	24.7 GB
Overall VBench ↑	0.8233	**0.8244**	0.7508	0.6943	0.7760	0.7901	0.8137	0.7958	0.7829

B.4 Multi-Prompt Long Video Generation



Figure 6: **Qualitative results of multi-prompt long video.** Our approach supports both global coherence via a shared prompt and fine-grained diversity via per-chunk local prompts.

Figure 6 shows multi-shot generation with a shared global prompt and per-chunk local prompts (Sec. 4.1): the global prompt keeps style and semantics consistent across scene transitions, while the local prompts supply per-chunk content.

B.5 ADDITIONAL QUALITATIVE RESULT

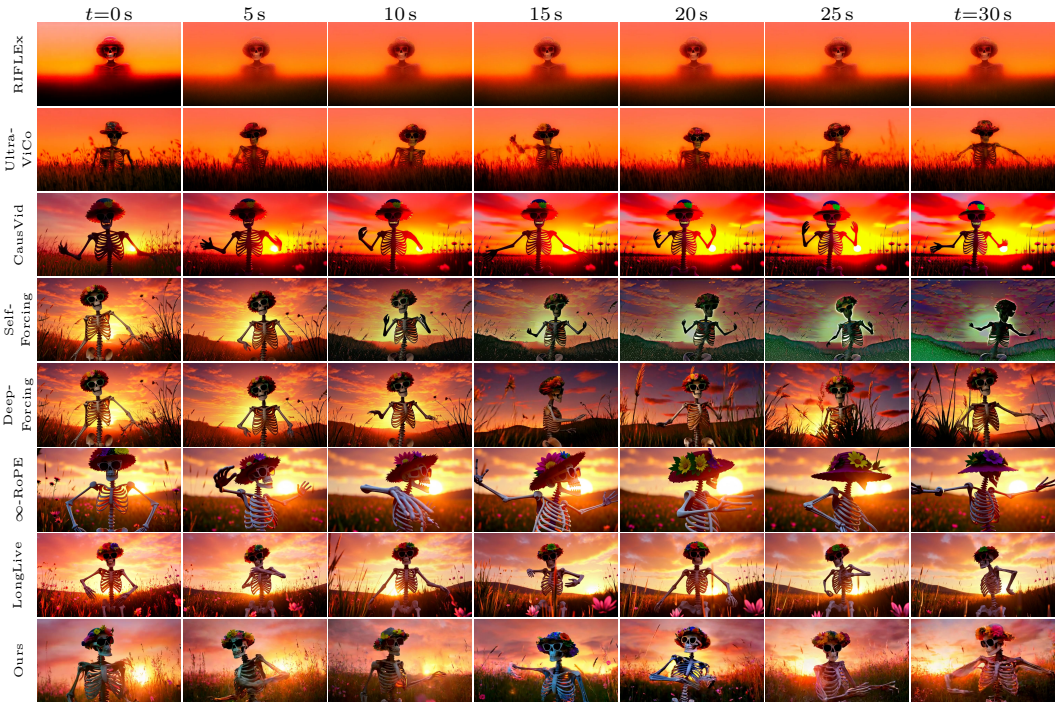


Figure 7: **Dense frame comparison.** Prompt: "a skeleton wearing a colorful flower hat and oversized sunglasses dancing wildly in a sunlit meadow at sunset". Frames show every 5,s over 30,s, with baseline model comparison.

To make temporal behavior over the full horizon visible, Figures 7–9 show frames sampled every 5 s from t=0 to t=30 s of the same 30s videos used in Table 1, for the bidirectional baselines (RIFLEx, UltraViCo), the autoregressive baselines (CausVid, Self-Forcing, Deep-Forcing, ∞-RoPE, LongLive), and FlowLong, all built on Wan 2.1-1.3B. Three failure modes recur across prompts. The bidirectional baselines stop moving within the first seconds and saturate in color or blur, so all seven frames look alike. The autoregressive baselines keep moving but drift: the background color shifts (Self-Forcing after about 15 s in Figure 7), the scene simplifies or the crowd disappears (Self-Forcing and Deep-Forcing in Figure 9), or the subject changes appearance (Self-Forcing in Figure 8), and the remaining motion repeats a small set of poses. FlowLong keeps the subject and the scene of the first frame through t=30 s while the pose, position, and camera framing keep changing, which is the behavior that the Dynamic Degree and the drift metrics of Table 2 measure. We provide full-length videos in the supplementary material.



Figure 8: **Dense frame comparison** Prompt: "a young male longboarder accelerating downhill, carving through tight turns on a scenic urban street". Frames show every 5,s over 30,s, with baseline model comparison.



Figure 9: **Dense frame comparison.** Prompt: "a kangaroo in a green polka-dot dress and sun hat strolling through the streets of Mumbai during a vibrant festival". Frames show every 5,s over 30,s, with baseline model comparison.

C Discussion: Fusion Space and Noise Re-injection

Why match in the clean space rather than in x_t. Because each chunk starts from independent noise, $\boldsymbol{x}_t^k \neq \boldsymbol{x}_t^{k+1}$ on the overlap even after synchronization begins. Averaging noisy latents, as in MultiDiffusion (Bar-Tal et al., 2023), therefore averages both the clean components and the independent noise components: the averaged noise has roughly $1/\sqrt{2}$ of the expected norm, so the fused latent leaves the thin shell on which $\boldsymbol{x}_t$ concentrates (Chung et al., 2022), and subsequent denoising runs on off-manifold inputs. Tweedie matching averages the denoised estimates *before* renoising, so the noise variance on the overlap is preserved exactly; under stochastic early-phase sampling this is the only fusion order that preserves it, since fusing after noise injection would again average independently injected noise. The x_t matching row of Table 3 isolates this effect with the sampler and schedule held fixed, and SyncTweedies (Kim et al., 2024a) reports the same ordering across fusion designs. For a deterministic affine sampler with a *shared* noisy latent, averaging clean estimates and averaging next-step latents coincide; that premise does not hold here.

D Additional Text-to-3DGS Results

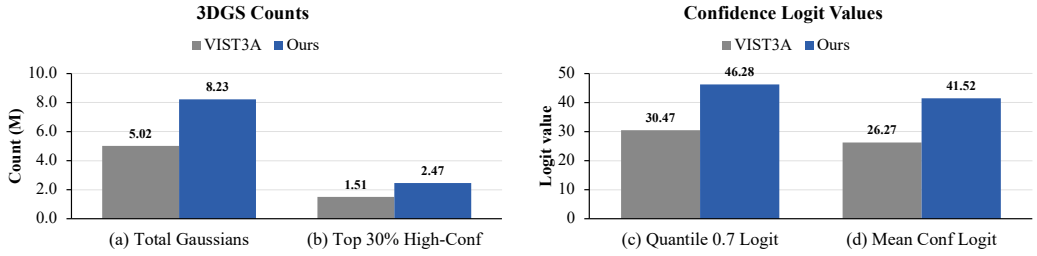


Figure 10: **Quantitative comparison of generated 3DGS.** (a) Total number of Gaussians produced per scene. (b) Number of Gaussians remaining after discarding the bottom 70% by confidence. (c) Average depth confidence logit across all Gaussians. (d) Confidence cutoff above which the top-30% most reliable Gaussians lie. All numbers are averaged over 100 SceneBench prompts.

AnySplat, the 3D Gaussian generator used in VIST3A, predicts a per-pixel depth confidence indicating how reliable the estimated geometry is; we use it to score the generated Gaussians. Our method generates 1.64× more Gaussians per scene than the baseline thanks to the longer video sequence, retains 2.47M Gaussians after keeping the top-30% by confidence, and raises the mean confidence logit from 26.27 to 41.52 and the 0.7-quantile logit from 30.47 to 46.28, i.e., more and higher-quality Gaussians across diverse scenes.