FlowBlending: Stage-Aware Multi-Model Sampling for Fast and High-Fidelity Video Generation

Jibin Song, Mingi Kwon, Jaeseok Jeong, Youngjung Uh*

Yonsei University

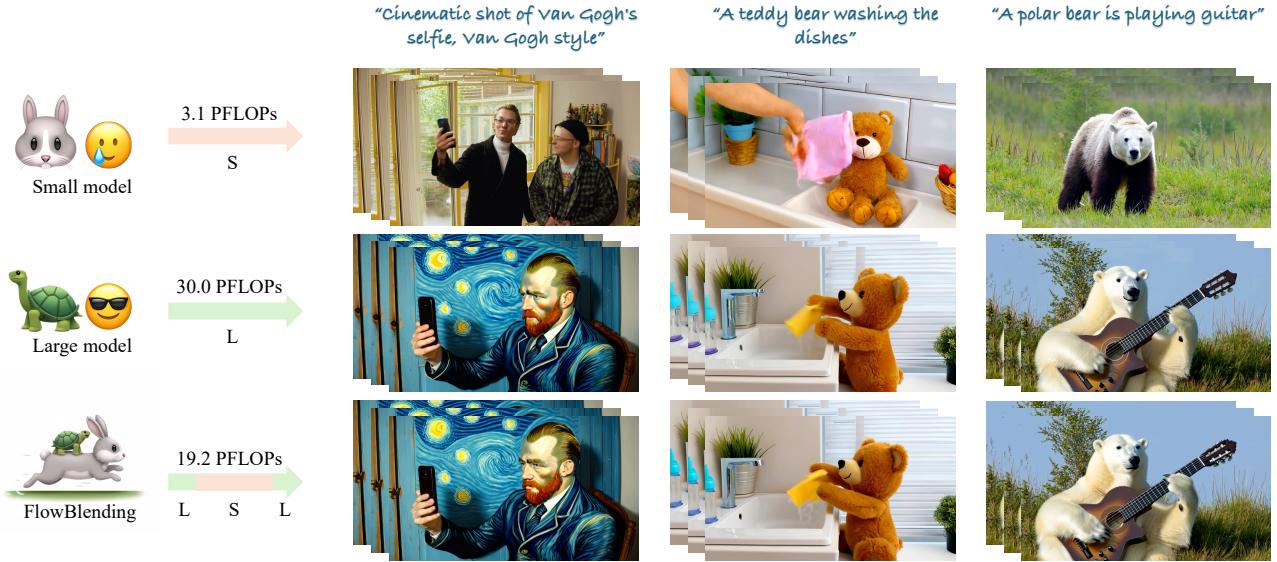{sjbpsh1, kwonmingi, jete_jeong, yj.uh}@yonsei.ac.kr

Figure 1. **Overview of FlowBlending.** The videos in each column are generated from the same initial noise and text prompt, but with different model allocation strategies. FlowBlending assigns a large model to process early and late denoising stages—establishing global structure and refining details, respectively— and assigns a small model to process intermediate denoising stages, where velocity divergence between the two models is minimal. This approach *preserves visual fidelity* of the large model while *reducing computation*.

Abstract

In this work, we show that the impact of model capacity varies across timesteps: it is crucial for the early and late stages but largely negligible during the intermediate stage. Accordingly, we propose FlowBlending, a stage-aware multi-model sampling strategy that employs a large model and a small model at capacity-sensitive stages and intermediate stages, respectively. We further introduce simple criteria to choose stage boundaries and provide a velocity-divergence analysis as an effective proxy for identifying capacity-sensitive regions. Across LTX-Video (2B/13B) and WAN 2.1 (1.3B/14B), FlowBlending achieves up to 1.65× faster inference with 57.35% fewer FLOPs, while maintaining the visual fidelity, temporal coherence, and semantic alignment of the large models. FlowBlending is also compatible with existing sampling-acceleration techniques, en-abling up to 2× additional speedup. Project page is available at: https://jibin86.github.io/flowblending_project_page

1. Introduction

Recent advances in diffusion-based video generation have significantly improved visual fidelity and temporal coherence [1, 2, 4, 6, 11, 14, 15, 18, 21, 42, 43]. However, these gains come with substantial computational cost due to the iterative denoising process and the increasing size of modern video diffusion models. While sampling acceleration has been extensively explored in text-to-image diffusion, progress on accelerating video diffusion has been limited.

Existing methods for accelerating diffusion sampling generally fall into two categories. One line of work accelerates sampling by sampling-step reduction algorithm, for example through improved numerical solvers or trajectory approximations [19, 28, 29, 32, 38]. Another direction

---

*Corresponding author.

1

distills the diffusion process into a smaller number of forward passes, aiming to match full-sampling performance with significantly fewer steps [7, 10, 16, 20, 23, 31, 35–37, 46, 48]. However, both strategies predominantly assume that *all timesteps require the same model capacity*, either by applying a single model uniformly across the denoising schedule or by compressing the entire model into a single distilled variant.

This leads to a fundamental question: *Do we truly need a large model for every diffusion step?* Notably, many recent video diffusion models are released in multiple capacity variants, e.g., LTX-Video[11] (2B / 13B) and WAN 2.1[42] (1.3B / 14B). Small models are significantly faster, but their visual quality is typically inferior, and they often fail to preserve semantic details. As illustrated in Figure 1, the WAN 2.1 small model (3.1 PFLOPs) struggles to accurately follow text prompts and often produces distorted or collapsed objects, despite its substantial efficiency advantage. In contrast, the large model, though computationally expensive, generates temporally coherent and semantically faithful videos. This disparity between large and small models highlights an opportunity: instead of uniformly applying high capacity across all timesteps, one may allocate the large model only where it provides the greatest benefit.

In this work, we make a key empirical observation inspired by the characteristics of the sampling process in video diffusion: **model capacity is not uniformly important across timesteps.** By evaluating the large and small models across multiple schedules, we find that: The early denoising stage influences global structure and motion, where the large model capacity yields noticeably better structure and semantic alignment than the small model. In addition, the late denoising stage refines high-frequency details and remove artifacts, again benefiting from the expressiveness of the larger model. In contrast, we demonstrate that the intermediate denoising stage admits substantial capacity reduction, as the small model yields outputs nearly identical to those of the large model.

These findings indicate that certain stages of the sampling process are substantially more capacity-sensitive than the others. Building on this insight, we propose FlowBlending, where the large model is used only in the early and late denoising stages, while the small model handles the majority of intermediate denoising stage. As shown in Figure 1, this approach preserves the visual quality of the large model while reducing computational cost a lot. Notably, our method requires no additional training, distillation, or architectural modification, and is complementary to existing acceleration techniques.

Furthermore, to identify when each model should be used, we introduce simple and practical strategies: semantic similarity between latents and a quantitative indicator of fine-detail quality. These strategies enable efficient capacity allocation while maintaining generation quality that is nearly indistinguishable from the large model.

In addition, we provide extensive experiments together with a velocity divergence analysis, which offers further insight into the sampling process and suggests a principled way to identify the early and late stage boundaries.

Across two open-source video diffusion models, LTX-Video (2B / 13B) and WAN 2.1 (1.3B / 14B), our stage-aware multi-model sampling achieves up to **1.65× faster** inference with **57.35% FLOPs** , while preserving the performance of the large model in visual fidelity, temporal coherence, and semantic alignment. In addition, our approach is orthogonal to existing acceleration methods, allowing up to an additional 50% FLOPs reduction when combined with complementary techniques.

2. Related work

Background: flow matching   Within the great success of diffusion models [13, 39], flow matching [24, 25, 40] has emerged as a widely adopted framework for modern generative modeling [9, 11, 22, 42]. Flow matching transfers the source distribution p_0 (e.g., Gaussian noise) to the target distribution p_1 (e.g., data distribution) by learning a velocity field $\mathbf{v}_t(\mathbf{x}; \theta)$ with a neural network θ. According to conditional flow matching (CFM), an intermediate latent $\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\mathbf{z}_1$ is formed at each timestep t and the network θ is trained using the optimal transport CFM (OT-CFM) loss:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, q(\mathbf{z}_1), p(\mathbf{z}_0)} \left\| \mathbf{v}_t \left((1 - t)\mathbf{z}_0 + t\,\mathbf{z}_1 \right) - (\mathbf{z}_1 - \mathbf{z}_0) \right\|^2. \tag{1}$$

The learned $\mathbf{v}_t(\cdot; \theta)$ progressively transfers the randomly sampled $z_0 \sim N(0, \mathbf{I})$ into z_1 by solving ordinary differential equation (ODE) with the number of function evaluations (NFE) where flow step $t \in [0, 1]$.

Acceleration of diffusion models   Diffusion models inherently suffer from high computational cost due to the large number of denoising steps required during sampling process. One line of work aims to mitigate this by employing more efficient numerical solvers, which substantially reduce the number of function evaluations (NFE) without additional training [19, 28, 29, 32, 38]. Another major direction focuses on step distillation [7, 10, 16, 20, 23, 31, 35–37, 46, 48], which compresses multi-step sampling into a smaller number of steps. Recently, several extensions have been proposed for video diffusion models as well [8, 44, 45, 49, 52], though these approaches typically require costly retraining. These two lines of research are largely orthogonal to our approach, as we focus on switching between models of different capacities during the sampling process rather than reducing the number of steps or retraining the model.

Multi-model sampling Yang et al. [47] showed that model capacity needs differ across timesteps in image diffusion, while Liu et al. [26] and Pan et al. [33] proposed mixing or allocating image models of varying sizes to different steps. However, their image-based analysis does not extend to video diffusion, where we find the opposite trend: *large models are crucial in the early stage to establish structure and coherent motion.*

3. Stage-aware multi-model sampling

A key challenge in accelerating video diffusion is to retain the generative performance of a high-capacity model while substantially reducing computational cost. Instead of modifying the sampling algorithm or retraining via distillation, we aim to reuse the existing large and small models as-is, allocating them across the sampling process. In other words, we frame this as a capacity allocation problem:

How can we preserve the quality of the large model while using the small model wherever possible?

Although the large model consistently yields higher fidelity, using it across all timesteps is prohibitively expensive. If we can identify the denoising timesteps where the small model's updates closely match those of the large model, we can safely replace the large model during those timesteps without degrading structure, motion dynamics, or identity consistency.

In the following sections, we analyze how model capacity contributes differently across the sampling process. In Section 3.1, we demonstrate that the early stage primarily governs global structure and coarse motion, where the expressive capacity of the model plays a critical role. Then, in Section 3.2, we show that the late stage is responsible for refining high-frequency details and resolving artifacts, again benefiting from the representational strength of the larger model. Finally, in Section 3.3, we introduce our method that leverages both characteristics.

3.1. Early structure formation

Our goal is to replace portions of the sampling process with the small model while retaining the large model's performance. We posit that key quality factors, such as visual fidelity, temporal coherence, and semantic alignment, are largely shaped in the early stage of the denoising process. We verify it in the video diffusion model via a simple but illustrative ablation.

Figure 2 compares four sampling schedules for WAN-2.1 (14B model as L, 1.3B as S): LLL (large-only), LSS (large→small), SLL (small→large), and SSS (small-only), where each letter denotes which model is used for a predefined segment of the timestep schedule.

As expected, LLL yields coherent global structure and strong alignment with the prompt. Remarkably, LSS, where the large model is used only during the earliest timesteps,
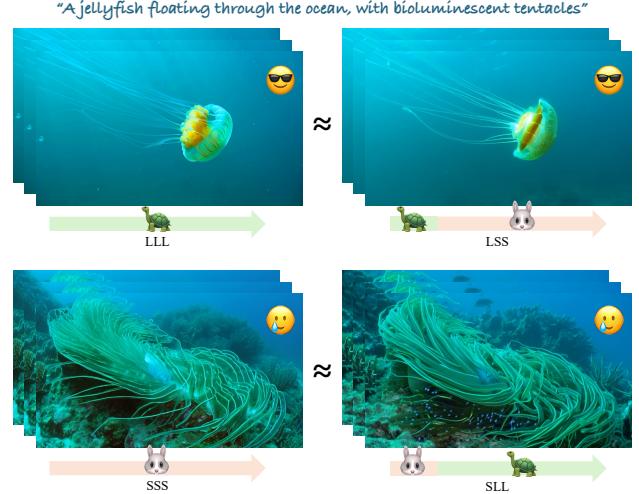

Figure 2. **Effect of model capacity during early denoising stage.** Comparison of WAN-2.1 sampling schedules (L = 14B, S = 1.3B). LSS (large only in early steps) closely matches LLL (large-only) in structure and motion, while SSS (small-only) exhibits temporal inconsistency and semantic misalignment. SLL (small only in early steps) likewise produces structure and motion patterns highly similar to SSS. This shows that the early stages are crucial for establishing global semantic and structural attributes.

produces nearly identical structure, motion coherence, and semantic consistency to LLL. In contrast, SSS fails to generate videos consistent with the text prompt, often producing drifting motion or incorrect object identity. Even SLL, which applies the large model after the initial steps, behaves similarly to SSS and struggles to recover correct semantics once the early structure formation has been misaligned.

These results indicate that **the early stage is capacity-sensitive**: The large model is critical for establishing coarse structure-level and motion-level attributes. Once the large model establishes the coarse structure, the small model can successfully take over the denoising process with minimal degradation in perceptual quality. We further provide the quantitative experiment in Section 4.2.

3.2. Late refinement

While LSS preserves global structure, motion, and semantics, we find that it may introduce subtle artifacts during the later denoising stage. These artifacts typically appear as spatial distortions or temporal flicker, which do not affect the overall structure but noticeably degrade perceptual quality.

Figure 3 illustrates this effect by comparing LLL (large-only), LSS (large→small), and LSL (large→small→large). Here, LSS uses the large model only during the early stage, while LSL reintroduces the large model exclusively in the final few timesteps.

As discussed in the previous subsection, LSS remains visually close to LLL in terms of coarse structure and mo-
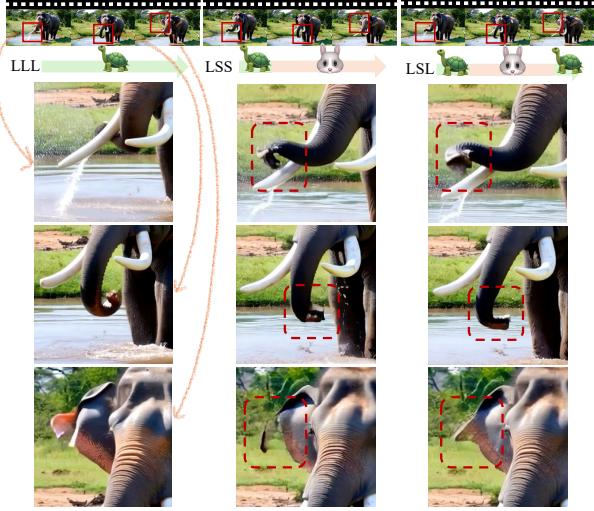
Figure 3. **Late stage ablation: LLL vs. LSS vs. LSL.** LSS preserves global structure similar to LLL but exhibits some artifacts. Reintroducing the large model only during the late stage (LSL) restores detail and reduces flicker, demonstrating that the late denoising stage is capacity-sensitive. Notably, the LSL schedule attains quality nearly indistinguishable from LLL while retaining the efficiency benefits of using the small model for most of the trajectory. Please zoom in to view the figures in detail.

tion. However, it fails to resolve high-frequency artifacts. In contrast, LSL successfully suppresses late stage distortions and restores fine detail. This indicates that the role of the large model in the late stage is not structure formation (which is already established early), but *artifact correction and detail refinement*. We therefore argue that **the late stage is capacity-sensitive once again**. We further provide the quantitative experiment in Section 4.3.

3.3. Stage-aware multi-model sampling

Based on these observations, we propose **FlowBlending (LSL)**, which allocates model capacity according to the sensitivity of each stage in the denoising process. The large model is used only during the early and late stages, where global structure formation and detail refinement are critical, while the small model handles the capacity-tolerant intermediate stage. This simple yet effective scheduling achieves near-large-model quality with reduced computation.

In the next section Section 4, we provide a quantitative evaluation of this strategy, analyzing its trade-offs across model blendings. We further describe how we determine the optimal boundaries of the early and late stages, inspired by the analysis between large and small models.

4. Experiment

4.1. Experimental setup

We evaluate the proposed sampling schedule on two representative open-source video diffusion models: LTX-Video

Schedule	DINO Sim↑	CLIP Sim↑	LPIPS↓	PSNR↑
LLL (Large-only)	100.00	100.00	-	-
LSS (Early-large, then small)	**95.74**	**96.97**	**2.76**	**24.30**
SLL (Early-small, then large)	65.58	81.10	30.49	12.62
SSS (Small-only)	65.01	80.87	30.54	12.59

Table 1. **Similarity to large-model baseline.** We report cosine similarity of DINO embeddings and CLIP embeddings, LPIPS, and PSNR averaged per-frame across the evaluation set. LSS remains close to LLL, indicating that early use of the large model preserves global semantics, while SSS diverges significantly.

[11] (2B / 13B) and WAN 2.1 [42] (1.3B / 14B). Evaluations are conducted on the PVD [3] and VBench [17]. We report FID [12] and FVD [41] using 284 generated samples, and four VBench metrics, Aesthetic Quality, Background Consistency, Subject Consistency, and Motion Smoothness, using 355 generated samples. We follow the default settings from each official repository. To quantify computational efficiency, we report the runtime and FLOPs of DiT blocks per generated video. Runtime for LTX-Video is measured on a NVIDIA A6000 GPU, and runtime for WAN 2.1 is measured on a NVIDIA A100 GPU. Please refer to Appendix A for more details.

4.2. Analysis of the early stage

In Section 3.1, we show that applying the large model only in the early stage is sufficient to establish the global structure and motion, after which the small model can take over the remaining denoising steps.

To quantitatively support the observation, we measure the similarity between each sampling schedule and the large-only baseline (LLL) using four metrics: (i) DINO [5] and CLIP [34] image-embedding similarity for semantic consistency, and (ii) LPIPS [50] and PSNR for low-level similarity, averaged across all frames and 355 generated videos. As shown in Table 1, LSS remains substantially closer to LLL across both semantic and low-level similarity metrics, confirming that invoking the large model only during the early stage is sufficient to preserve global structure and text-aligned semantics. In contrast, both SLL and SSS exhibit significantly lower similarity. Notably, SLL performs comparably to SSS despite using the large model in the latter part of sampling. This result indicates that if the early structure formation is misaligned, the subsequent steps cannot recover the global structure or the semantic alignment to the prompt, even when a high-capacity model is used later in the denoising process.

These findings suggest that **the early stage is capacity-sensitive**, where employing the large model is crucial for establishing stable global structure and semantic alignment. For full experimental details, please refer to Appendix B.1.

Boundary of the early stage Figure 4 extends the DINO-similarity experiment by varying the early stage boundary at
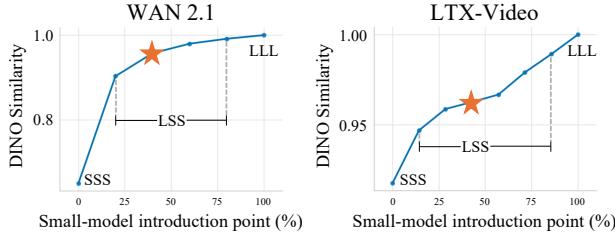
Figure 4. **DINO-based identification of the early stage boundary.** We measure the frame-wise DINO similarity between schedules that switch from the large to the small model ($L \to S$) at different early stage boundaries and the large-only baseline (LLL). A sharp decline in similarity emerges beyond a specific point in the curve. Boundaries chosen just before this drop (typically above $\sim 96\%$ similarity) preserve global structure and motion.

which the schedule transitions from the large model to the small model ($L \to S$ on the LSS schedule). We first observe that the DINO similarity curve exhibits a sharp drop after a certain point; before this drop, the similarity to the large-only baseline remains consistently high. Motivated by this structure in the graph, we select the early-stage boundary just before the curve begins to decline. Empirically, all such candidates fall above roughly 96% similarity, and schedules using these boundaries generate videos whose global structure and motion are nearly indistinguishable from the large-only baseline. This indicates that maintaining similarity above this threshold is sufficient to preserve high-level fidelity while enabling acceleration.

4.3. Analysis of the late stage

In Section 3.2, we show that applying the large model at the late stage refines details and reduces artifacts. Table 2 quantitatively supports it through FID results: The LSL schedule consistently achieves lower FID than LSS, indicating that late stage large-model updates improve visual fidelity.

Taken together, these results lead to our core principle: *Use the large model early to establish global structure, and use it again at the end to refine details.*

Boundary of the late stage. We observe that the FID curve exhibits a V-shape in Figure 5. Interestingly, for both WAN and LTX-Video, choosing an appropriate late stage boundary consistently lowers the FID. Moreover, when the early stage boundary is properly selected, all LSL schedules produce outcomes that closely match those of the large-only baseline (LLL).

A notable finding is that incorporating the small model during the intermediate stage can make the generated video appear more similar to real footage. While LLL produces high-quality results overall, it occasionally yields overly smooth surfaces, whereas well-configured LSL schedules generate objects with more natural and realistic textures. We speculate that a moderate degree of "noisiness" introduced

Figure 5. **FID-based identification of the late stage boundary.** We fix the early stage boundary identified in Section 4.2 and vary only the late stage boundary. The resulting FID curve exhibits a V-shape, where the minimum corresponds to the optimal late stage boundary used in our LSL schedule. This trend consistently appears in both WAN and LTX-Video, demonstrating that a properly chosen late boundary yields the best sweet spot with detail refinement and artifact suppression.

by the small model may contribute to this increased realism.

However, excessive reliance on the small model leads to artifacts or temporal flicker, and the FID curve increases once the proportion of small model usage surpasses a certain threshold. The observed V-shape FID plot in Figure 5 reflects this behavior. Our experiments reveal a clear sweet spot: moderate use of the small model during the intermediate stage, followed by large model refinement at the end, enhances fine-grained texture without introducing artifacts. We further argue that this sweet spot can be identified through artifact-sensitive and pixel-level detail metrics such as FID. Additional qualitative results are provided in Appendix B.2.

4.4. Quantitative results

Table 2 summarizes the quantitative results in schedules across two models. Our proposed **LSL** schedule achieves FID, FVD, Aesthetic, Background, Subject, and Motion scores that are nearly indistinguishable from the large-only baseline (LLL), while accelerating inference by up to $1.65\times$ with 57.35% FLOPs. Compared to the small-only schedule (SSS), LSL clearly preserves the performance characteristics of LLL, demonstrating that selectively invoking the large model at key stages is sufficient to maintain overall quality.

In contrast, the early-large schedule (LSS) successfully preserves global structure but fails to resolve high-frequency artifacts, resulting in consistently lower scores across several metrics. This behavior aligns with our analysis in Figure 3, where late stage model capacity is shown to be critical for detail refinement.

These results reinforce the central claim of this work: **Video quality does not require high model capacity at every step; invoking the large model only at structurally and detail-critical stages preserves fidelity while reducing computation.**

5

Table 2. **Comparison of video quality and efficiency across sampling schedules.** LSL achieves quality comparable to the large-only baseline (LLL), while reducing runtime and FLOPs. LSS preserves global structure but leaves late stage artifacts, whereas SSS degrades across most metrics.

Schedule	FID ↓	FVD ↓	Aesthetic ↑	Background ↑	Subject ↑	Motion ↑	Runtime ↓	TFLOPs ↓
LLL (Large-only)	<u>5.73</u>	834.26	**45.43**	93.79	**90.96**	97.56	$49.73_{\pm 0.31}$	3496
LSL (Stage-aware, Ours)	**5.70**	**752.07**	<u>44.52</u>	**93.96**	<u>90.79</u>	**97.74**	$30.18_{\pm 0.04}$	2005
LSS (Early-large)	5.75	<u>759.39</u>	44.25	<u>93.89</u>	90.66	<u>97.63</u>	$25.44_{\pm 0.07}$	1632
SSS (Small-only)	6.28	951.79	43.14	91.70	86.23	96.02	$10.69_{\pm 0.01}$	514

(a) **LTX-Video**. Runtime is measured on a NVIDIA A6000 GPU.

Schedule	FID ↓	FVD ↓	Aesthetic ↑	Background ↑	Subject ↑	Motion ↑	Runtime ↓	TFLOPs ↓
LLL (Large-only)	7.55	2,831.94	**57.56**	<u>96.21</u>	<u>94.61</u>	<u>98.10</u>	$653.12_{\pm 0.80}$	29950
LSL (Stage-aware, Ours)	**7.48**	<u>2,752.17</u>	<u>57.42</u>	**96.29**	**94.61**	**98.11**	$439.36_{\pm 0.56}$	19222
LSS (Early-large)	<u>7.51</u>	2,803.48	57.27	96.11	94.47	98.10	$335.20_{\pm 0.44}$	13858
SSS (Small-only)	7.55	**2,556.55**	56.21	95.71	94.15	97.93	$124.05_{\pm 0.47}$	3129

(b) **WAN 2.1**. Runtime is measured on a NVIDIA A100 GPU.

5. Extended analyses

In Section 5.1, we examine the distribution of velocity divergence between the large and small model velocities and analyze how its behavior aligns with the identified boundary regions. In Section 5.2, we validate the proposed stage-aware schedule across a wide range of configurations to demonstrate that our boundary choice is appropriate.

5.1. Velocity divergence across sampling steps

A natural way to determine when to use the large or small model is to measure how differently they update the latent state during denoising. To this end, we compare the velocity fields predicted by the large and small models at each timestep t. Given a latent $\mathbf{z}_t$, let $\mathbf{v}_t^{(L)}$ and $\mathbf{v}_t^{(S)}$ denote the velocities predicted by the large (L) and small (S) models, respectively. We compute two metrics: $\text{cosine\_dist}(t) = 1 - \cos\left(\mathbf{v}_t^{(L)}, \mathbf{v}_t^{(S)}\right)$, $\ell_2\text{\_dist}(t) = \|\mathbf{v}_t^{(L)} - \mathbf{v}_t^{(S)}\|_2$.

We estimate these quantities by sampling from random noise using the large model, while feeding the same intermediate latents through the small model. As shown in Figure 6, the divergence curve consistently follows a U-shaped pattern across the sampling process. The intermediate timesteps show low divergence, indicating that the small model and large model predict nearly identical update directions in this region. In contrast:

- In the **early stage**, we observe large variance across samples, even though the mean divergence remains moderate. This variance reflects the instability of the small model's predictions. Because this stage governs global structure and motion formation, even small deviations can drastically alter the resulting layout. Consequently, employing the large model at this stage plays a crucial role in generating high-quality videos.
- In the **late stage**, the mean divergence is higher than in the

Figure 6. **Velocity divergence across diffusion steps.** We compute cosine and ℓ_2 distances between velocities $\mathbf{v}_t^{(L)}$ and $\mathbf{v}_t^{(S)}$ across timesteps. Divergence is lowest in the intermediate stage, indicating that the small model can reliably denoise there. Early stage exhibits high variance (structure formation), while late stage shows high mean divergence (detail refinement). This supports allocating the large model to the early and late stages.

intermediate stage, indicating that the velocity predictions of the small and large models differ most in this stage. This stage corresponds to fine-grained detail refinement and artifact suppression, both of which critically rely on the representational capacity of the large model.

As shown in Figure 6, we plot the velocity divergence between the small and large models. The empirically determined stage boundaries from Section 4 are marked with red arrows. Interestingly, for both LTX-Video and WAN 2.1, the stage boundaries empirically determined in Section 4 coincide with the boundaries inferred from the cosine distance curves between the small and large models. The same pattern holds for the ℓ_2 distance. These observations suggest that the characteristic U-shape of velocity divergence could offer a *convenient proxy for identifying stage boundaries*. In particular, we speculate that the onset of increasing vari-

Figure 7. **FID vs. FLOPs across schedule configurations.** Schedules beginning with S (SXX) show significantly degraded quality. Schedules using the large model in the early stage (LXX) achieve substantially better performance. Our selected early and late boundaries (yellow dots), and especially the full stage-aware schedule (yellow star), lie on the Pareto frontier, achieving LLL-level quality with lower FLOPs.

ance in the late stage corresponds to the late stage boundary. Notably, this trend does not appear to be model-specific, metric-specific, or condition-specific. Even unconditioned velocities exhibit a similar behavior (Appendix C).

Taken together, these findings suggest a simple underlying principle of the LSL schedule introduced earlier: **Use the large model when divergence is high and the small model when divergence is low.**

5.2. Extensive schedule comparison

We evaluate nearly the entire space of possible schedules. To do so, we divide the sampling trajectory into 12.5% segments and generate combinations of large (L) and small (S) model assignments. This covers LLL, LSS, LSL, and SSS, along with a broad set of additional S-starting schedules (denoted as SXX in Figure 7) for comparison. For each schedule, we measur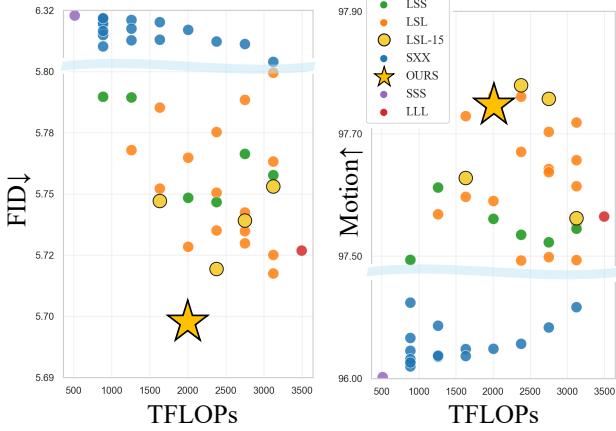e the FID and Motion Smoothness of generated videos. Figure 7 plots FID and Motion Smoothness against computational cost (FLOPs) for the entire schedule configurations.

Consistent with our earlier observations, all schedules that begin with S (blue points in Figure 7) yield significantly worse FID and Motion Smoothness than those beginning with L. Surprisingly, even the schedule that applies the large model only once at the very beginning (the leftmost green point) performs better than a schedule that applies the large model everywhere except the first segment (the rightmost blue point). This highlights the crucial role of the early stage in establishing overall video quality.

In Figure 7, the yellow points correspond to LSL schedules that adopt the early stage boundary identified

in Section 4.2. They generally achieve a more favorable FID–FLOPs trade-off than the orange LSL schedules with alternative early stage boundaries, indirectly supporting the superiority of our boundary choice in Section 4.2.

Among these variants, our proposed LSL schedule (marked with a star) attains the lowest FID while preserving an excellent FID–FLOPs trade-off. Despite much lower compute, it delivers consistently strong performance, and qualitatively, its outputs remain nearly indistinguishable from those produced by the large-only baseline (LLL).

Figure 7 reveals two additional observations about boundary sensitivity. (i) Shifted variants, which are the orange points aligned vertically with the yellow star match our FLOPs budget but shift the early or late boundary. Across both FID and motion smoothness, our configuration (yellow star) consistently outperforms these shifted LSL variants (orange points aligned vertically).

(ii) Expanded/reduced variants, shown as the orange points immediately to the left or right of the yellow star, add or remove a single boundary segment. These variants (orange points in the vertical band around the yellow star) also yield worse FID and motion smoothness compared to our configuration (yellow star), indicating that even small deviations from the selected boundaries degrade quality.

Together, these results show that, across a wide range of possible configurations, the identified early and late boundaries form a robust sweet spot for capacity allocation. All quantitative metrics exhibit consistent trends; full tables are included in Appendix D.

6. Compatibility with other acceleration methods

In this section, we show that our approach is orthogonal to existing acceleration methods: (a) sampling-step reduction algorithm and (b) distillation-based model training.

(a) Compatibility with sampling-step reduction Algorithm. DPM++ accelerates the denoising process by reducing the number of function evaluations (NFE). We evaluate our LSL schedule together with DPM++ [29] with the NFE reduced by half. As illustrated in Figure 8, even with the reduced NFE, LSL reproduces results comparable to the large-only baseline (LLL). In contrast, the small-only schedule (SSS) produces noticeable artifacts: the fork and the pastry deform (left example), and abrupt scene transitions (right example). This demonstrates that our findings remain valid when combined with DPM++. Quantitatively, Table 3 shows that DPM++ with LSL reduces total TFLOPs by about $2\times$, with negligible quality loss.

(b) Compatibility with the distilled model. Next, we replace the small model (S) in our LSL schedule with the
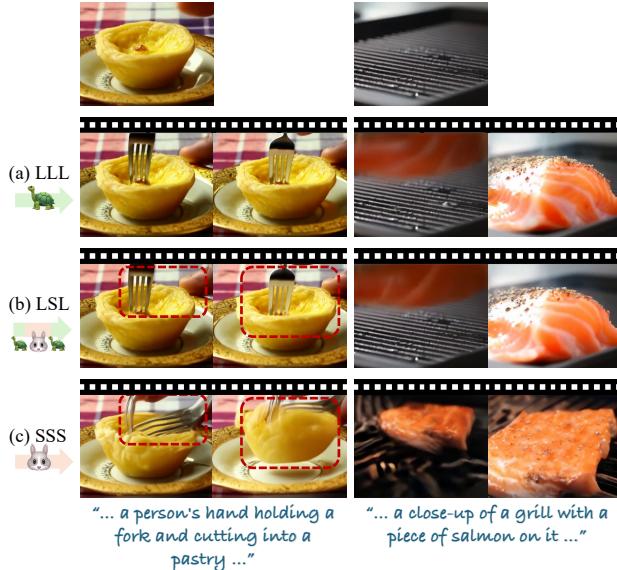
Figure 8. **Compatibility with DPM++ solver.** Ours (LSL) is compatible with DPM++ solvers, reproducing similar videos to the videos using only the large model (LLL). Please zoom in to view the figures in detail.

Setting	Solver	Schedule	NFE	FID ↓	FVD ↓	TFLOPs ↓
Sampling-Step Reduction	DPM++	LLL	20	<u>5.78</u>	**769.83**	1748
		LSL	20	**5.77**	<u>773.32</u>	928
		SSS	20	6.11	1,023.65	257
Distillation	ODE	LLL	40	**5.73**	**834.26**	3496
		LDL	24	<u>5.93</u>	<u>1,104.18</u>	1774
		DDD	8	6.50	1,843.53	51

Table 3. **Compatibility with existing acceleration methods.** Our stage-aware schedule maintains large-model quality even when combined with reduced-step solvers or distilled backbones, demonstrating orthogonality to existing accelerators.

officially released distilled variant (D) of LTX-Video. The distilled model was trained to generate videos in just eight sampling steps. Within the LSL framework, we employ it only for the intermediate stage, corresponding to four sampling steps, and denote this configuration as **LDL**. As shown in Figure 9, LDL reproduces results comparable to the large-only baseline (LLL), avoiding issues such as sudden subtitle appears (left) or hand deformation artifacts (right) that appear in SSS. Table 3 further confirms that LDL reduces total FLOPs to nearly half of LLL while maintaining comparable perceptual quality. These results confirm that ours complements existing acceleration techniques and can be *integrated* with them for better efficiency.

7. Discussion and conclusion

We investigate how model capacity contributes differently across the video diffusion denoising process. Through empirical analyses, we show that the importance of model capacity varies across denoising stages: The early stage gov-

Figure 9. **Compatibility with step distillation model.** The step distillation model (D) with small capacity can replace the original small model (S). As done with the original small model, LDL reproduces the results with LLL. In contrast, DDD does not. Please zoom in to view the figures in detail.

erns global structure and motion, and the late stage refines high-frequency details and suppresses artifacts, both of which are capacity-sensitive.

Building on this insight, we introduce FlowBlending, a stage-aware multi-model sampling strategy that allocates the large model to the early and late stages while delegating the intermediate stage to the small model. FlowBlending preserves the visual fidelity, temporal coherence, and semantic alignment of the large model, yet notably reduces sampling cost. Across LTX-Video and WAN 2.1, FlowBlending achieves up to **1.65× faster** inference with **57.35% FLOPs** while not compromising the video quality.

To determine where model capacity matters most, we proposed a practical method for identifying the early and late stage boundaries using DINO similarity and FID-based trade-off analysis. Interestingly, these empirically chosen boundaries coincide with regions of increasing variance in the velocity divergence between the large and small models.

With the analysis, we showed that FlowBlending is orthogonal to existing acceleration techniques. In particular, we empirically observe that DPM++ solver produces outputs that more closely match those of the large model. It is another promising research avenue to determine whether improved solvers shift the effective stage boundaries.

While FlowBlending is broadly applicable, a key limitation remains: stage boundaries often need to be re-estimated when the diffusion model changes. Automatic boundary detection or model-agnostic stage criteria would further improve the usability and generality of stage-aware sampling.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1

[2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[3] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025. 4, 1, 5

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

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 4

[6] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023. 1

[7] Trung Dao, Thuan Hoang Nguyen, Thanh Le, Duc Vu, Khoi Nguyen, Cuong Pham, and Anh Tran. Swiftbrush v2: Make your one-step diffusion model better than its teacher. In *European Conference on Computer Vision*, pages 176–192. Springer, 2024. 2

[8] Hangliang Ding, Dacheng Li, Runlong Su, Peiyuan Zhang, Zhijie Deng, Ion Stoica, and Hao Zhang. Efficient-vdit: Efficient video diffusion transformers with attention tile. *arXiv preprint arXiv:2502.06155*, 2025. 2

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 2

[10] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024. 2

[11] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024. 1, 2, 4, 5

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 4, 1

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 1

[15] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 1

[16] Yi-Ting Hsiao, Siavash Khodadadeh, Kevin Duarte, Wei-An Lin, Hui Qu, Mingi Kwon, and Ratheesh Kalarot. Plug-and-play diffusion distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13743–13752, 2024. 2

[17] 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*, pages 21807–21818, 2024. 4, 1, 5

[18] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024. 1

[19] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 1, 2

[20] Beomsu Kim, Yu-Guan Hsieh, Michal Klein, Marco Cuturi, Jong Chul Ye, Bahjat Kawar, and James Thornton. Simple reflow: Improved techniques for fast flow models. *arXiv preprint arXiv:2410.07815*, 2024. 2

[21] Mingi Kwon, Seoung Wug Oh, Yang Zhou, Difan Liu, Joon-Young Lee, Haoran Cai, Baqiao Liu, Feng Liu, and Youngjung Uh. Harivo: Harnessing text-to-image models for video generation. In *European Conference on Computer Vision*, pages 19–36. Springer, 2024. 1

[22] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 2

[23] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36:20662–20678, 2023. 2

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[25] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David

Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. 2

[26] Enshu Liu, Xuefei Ning, Zinan Lin, Huazhong Yang, and Yu Wang. Oms-dpm: Optimizing the model schedule for diffusion probabilistic models. In *International Conference on Machine Learning*, pages 21915–21936. PMLR, 2023. 3

[27] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7353–7363, 2025. 6

[28] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022. 1, 2

[29] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, 2025. 1, 2, 7

[30] Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K Wong. Fastercache: Training-free video diffusion model acceleration with high quality. *arXiv preprint arXiv:2410.19355*, 2024. 6

[31] Kangfu Mei, Mauricio Delbracio, Hossein Talebi, Zhengzhong Tu, Vishal M Patel, and Peyman Milanfar. Codi: Conditional diffusion distillation for higher-fidelity and faster image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9048–9058, 2024. 2

[32] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1, 2

[33] Zizheng Pan, Bohan Zhuang, De-An Huang, Weili Nie, Zhiding Yu, Chaowei Xiao, Jianfei Cai, and Anima Anandkumar. T-stitch: Accelerating sampling in pre-trained diffusion models with trajectory stitching. *arXiv preprint arXiv:2402.14167*, 2024. 3, 6

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 4

[35] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2

[36] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 6

[37] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024. 2, 6

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2

[39] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. 2

[40] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024. 2

[41] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. In *arXiv*, 2019. 4, 1

[42] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 2, 4, 5

[43] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report, 2023. 1

[44] Yushu Wu, Yanyu Li, Anil Kag, Ivan Skorokhodov, Willi Menapace, Ke Ma, Arpit Sahni, Ju Hu, Aliaksandr Siarohin, Dhritiman Sagar, et al. Taming diffusion transformer for efficient mobile video generation in seconds. *arXiv preprint arXiv:2507.13343*, 2025. 2

[45] Yushu Wu, Zhixing Zhang, Yanyu Li, Yanwu Xu, Anil Kag, Yang Sui, Huseyin Coskun, Ke Ma, Aleksei Lebedev, Ju Hu, et al. Snapgen-v: Generating a five-second video within five seconds on a mobile device. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2479–2490, 2025. 2

[46] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8196–8206, 2024. 2, 6

[47] Shuai Yang, Yukang Chen, Luozhou Wang, Shu Liu, and Yingcong Chen. Denoising diffusion step-aware models. *arXiv preprint arXiv:2310.03337*, 2023. 3

[48] 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*, pages 6613–6623, 2024. 2

[49] Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. Accvideo: Accelerating video diffusion model with synthetic dataset. *arXiv preprint arXiv:2503.19462*, 2025. 2

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4

[51] Yuechen Zhang, Jinbo Xing, Bin Xia, Shaoteng Liu, Bohao Peng, Xin Tao, Pengfei Wan, Eric Lo, and Jiaya Jia.

Training-free efficient video generation via dynamic token carving. *arXiv preprint arXiv:2505.16864*, 2025. 6

[52] Zhixing Zhang, Yanyu Li, Yushu Wu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Dimitris Metaxas, Sergey Tulyakov, et al. Sf-v: Single forward video generation model. *Advances in Neural Information Processing Systems*, 37:103599–103618, 2024. 2

FlowBlending: Stage-Aware Multi-Model Sampling for Fast and High-Fidelity Video Generation

Supplementary Material

A. Experimental Details

We evaluate the proposed sampling schedule on two representative open-source video diffusion models: LTX-Video (2B / 13B) [11] and WAN 2.1 (1.3B / 14B) [42]. LTX-Video generates 65-frame videos at a resolution of $608{\times}342$ and 30 fps on the VBench dataset [17], and $832{\times}480$ videos on the PVD [3]. WAN 2.1 generates 61-frame videos at $832{\times}480$ and 16 fps across both datasets.

For quantitative evaluation, we report FID [12] and FVD [41] on 284 generated samples using subset of PVD prompts, and VBench metrics on 355 generated samples using VBench dataset prompts or 284 generated samples same as FID and FVD. We use the default evaluation configurations provided by each official repository: LTX-Video experiments adopt the ODE sampler, while WAN 2.1 experiments follow the UniPC sampler.

To assess computational efficiency, we additionally measure runtime and compute FLOPs of DiT blocks per generated video. LTX-Video runtime is measured on an NVIDIA A6000 GPU, and WAN 2.1 runtime is measured on an NVIDIA A100 GPU.

Most of our quantitative results, including FID, FVD, and qualitative comparisons, are obtained using videos generated on the PE Video Dataset, which we adopt as our primary evaluation set. Although we also report VBench metrics, the results are rely on PVD due to its richer prompts and better alignment with the types of video quality differences we aim to study.

VBench includes a large collection of short and simple prompts designed to probe a wide range of video properties. However, many of these prompts produce videos that do not align with the types of content we aim to evaluate. For instance, prompts such as "a leafless tree", "young guy with VR headset", or "milk and cinnamon rolls" typically result in highly static scenes with minimal motion or structural complexity. Such samples make it difficult for VBench metrics to capture the aspects we care about most object consistency, motion coherence, and subtle high-frequency artifacts.

In contrast, the PE Video Dataset provides substantially richer text–video pairs with more dynamic and semantically meaningful prompts (e.g., "a dog running across a snowy field,"). Evaluations with these prompts better reflect the perceptual differences we observe empirically across sampling schedules. While both VBench and PE yield broadly similar trends, PE offers a closer match to our practical experience regarding model quality, making it our primary benchmark dataset.

B. More ablation

B.1. Early stage ablation

Experimental details To determine the early stage boundary, we conduct additional ablation experiments using LTX-Video (2B / 13B), an image-to-video model, and WAN 2.1 (1.3B / 14B), a text-to-video model. We follow the default configurations provided by the official repositories, including the sampler, noise schedule, and total number of denoising steps. In detail, we use 40 timesteps for LTX-Video and 50 timesteps for WAN 2.1.

For evaluation, we generate 355 videos using the text and image prompts from the VBench Dataset [17]. To quantify how well the small model can replace the large model at different early stages, we measure frame-wise DINO similarity, CLIP similarity, LPIPS, PSNR, and SSIM between each hybrid schedule that switches from the large to the small model ($L \rightarrow S$) at a given timestep and the large-only baseline (LLL). Each metric is computed between corresponding frames of the hybrid schedule and the LLL output, and then averaged across all frames and videos:

$$\text{Metric}(t_{\text{switch}}) = \frac{1}{N} \sum_{i=1}^{N} d\big(x_i^{(L \rightarrow S)}, \ x_i^{(L)}\big), \tag{2}$$

where $x_i^{(L)}$ denotes the i-th frame of the large-only baseline.

We progressively shift the early stage boundary across timesteps: LTX-Video experiments vary the boundary in increments of 5 timesteps, and WAN 2.1 experiments vary it in increments of 10 timesteps. This allows us to precisely characterize how sensitive the early denoising stage is to model capacity across both architectures.

Figure A1. **LTX-Video: Experiments for choosing early stage boundary** . The star sign represents our choice of the early stage boundary. The X-axis is the small model introduction point (%), where the small model follows the large model. We present the results for CLIP, LPIPs, PSNR, and SSIM in order.

Figure A2. **WAN 2.1: Experiments for choosing early stage boundary.** The star sign represents our choice of the early stage boundary. The X-axis is the small model introduction point (%), where the small model follows the large model. We present the results for CLIP, LPIPs, PSNR, and SSIM in order.

Results Figure A1 and Figure A2 plot the early stage boundary on the x-axis, expressed as the Small-model introduction point (%) measured from the start of the sampling steps. The y-axis reports the similarity between the outputs of the large-only model and those of the LSS schedule, evaluated using DINO, CLIP, LPIPS, PSNR, and SSIM. These curves indicate how closely the results resemble the large-only baseline as the early stage boundary varies. As shown in the figures, not only the DINO similarity discussed in the main text but all metrics exhibit a consistent trend: a sharp decline in similarity emerges beyond a specific point in the curve. We select the boundary just before this drop, which preserves global structure and motion.

Among all metrics, we find that DINO and CLIP similarity, both of which capture richer semantic information, align most closely with our observations. This is likely because small pixel-level differences measured by metrics such as LPIPS, PSNR, or SSIM do not always correspond to perceptually meaningful changes.

More qualitative results Please see the attached HTML file ("index.html") for the ablation study on selecting the early-stage boundary. We provide videos generated with different small-model introduction points (%) in the denoising process, as illustrated in Figure 4. A setting of 40% (LSL) means that the small model is introduced starting from 40% of the denoising process, and 100% (LLL) corresponds to not introducing the small model at all. When the introduction point is delayed up to 40%, the resulting appearance and motion remain nearly identical to the large-only model (LLL). However, pushing the boundary further to 60% (LSS) leads to a different motion. Therefore, our final choice, introducing the small model at 40%, provides the optimal balance between quality and efficiency.

B.2. Late stage ablation

Experimental details This section provides additional information on the setup used to determine the late stage boundary. We experiment with two video diffusion families: LTX-Video (2B/13B), an image-to-video model, and WAN 2.1 (1.3B/13B), a text-to-video model, and generate 284 videos using the PVD dataset for evaluation. We follow all default settings from each official repository, including the sampler configuration and the total number of timesteps, where LTX-Video uses 40 steps and WAN 2.1 uses 50 steps. In all experiments, we fix the early stage boundary identified in Section 4.2 and vary only the late stage boundary. For each schedule, we measure performance using VBench metrics—Subject Consistency, Background Consistency, Temporal Flickering, Motion Smoothness, Dynamic Degree, Aesthetic Quality, and Image Quality—along with

---
Algorithm 1 How to find early stage boundary via relative slope threshold
---
1: **Input:** The small model introduction points (%) $\{x_i\}_{i=0}^{N-1}$, DINO similarities $\{y_i\}_{i=0}^{N-1}$, threshold $\alpha \in (0,1)$
2: **Output:** Early stage boundary index k, Early stage boundary location (x_k, y_k)
3: **Step 1: Compute local slopes**
4: **for** $i = 0$ to $N - 2$ **do**
5:     $\Delta x \leftarrow x_{i+1} - x_i$
6:     **if** $\Delta x = 0$ **then**
7:         $s_i \leftarrow 0$
8:     **else**
9:         $s_i \leftarrow (y_{i+1} - y_i)/\Delta x$
10:     **end if**
11: **end for**
12: **Step 2: Reference slope**
13: $s_{\text{ref}} \leftarrow s_0$
14: **Step 3: Find index where slope falls below threshold**
15: $k \leftarrow N - 1$
16: **for** $i = 0$ to $N - 2$ **do**
17:     **if** $s_i \leq \alpha\, s_{\text{ref}}$ **then**
18:         $k \leftarrow i + 1$
19:         **break**
20:     **end if**
21: **end for**
22:
23: **return** (k, x_k, y_k)

FID and FVD. Ablations are performed by gradually shifting the late stage boundary forward from the fixed early boundary, using 5-timestep intervals for LTX-Video and 10-timestep intervals for WAN 2.1. All LTX-Video results are obtained on an NVIDIA A6000 Ada GPU, while WAN 2.1 evaluations are conducted on an NVIDIA A100 GPU due to model size differences.

Results In Figure A3, we provide the results of LTX-Video with the Vbench metrics, FID, and FVD. The star sign represents our late boundary choice, and the optimal choice of the late boundary enables us to maintain the video quality. As shown in the Figure A3, FID and FVD graphs have the best scores with our late boundary choice. With our late boundary choice, the video quality-related metrics and background consistency also show the best and comparable results, respectively. Likewise, the temporal flickering and motion smoothness achieves nearly on par with the best results. Although our late boundary choice always work on the other metrics, their measurements are far from our purpose of using the big model at the late stage. For example, measuring the dynamic degree and aesthetic score is irrelevant to maintaining fine details of the video. In the imaging quality metric, using more big model results in a better score. However, the observed visual difference is negligible.

In Figure A4, we provide the Vbench metrics, FID, and FVD. Similar to the LTX-Video setting, we can identify an optimal late boundary by locating the points where FID and FVD are minimized. Moreover, the overall trends are consistent with those observed for LTX-Video across different metrics, indicating that our late boundary strategy behaves robustly regardless of the specific evaluation metric.

In Algorithm 1, we provide the pseudo-algorithm to find the early stage boundary with the slope. We set the threshold to find the late stage boundary with FID.

Qualitative results Please see the attached HTML file ("index.html") for the ablation study on selecting the late-stage boundary. Therefore, reintroducing the large model at 20%, our final choice, offers the optimal balance between video quality and efficiency. We provide videos generated with different large-model reintroduction points (% from the end) in the denoising process, as illustrated in Figure 5. A setting of 40% (LSL) means that the large model is reintroduced starting from the last 40% of the denoising trajectory; 20% (LSL-ours) reintroduces the large model from the last 20%; and 0%
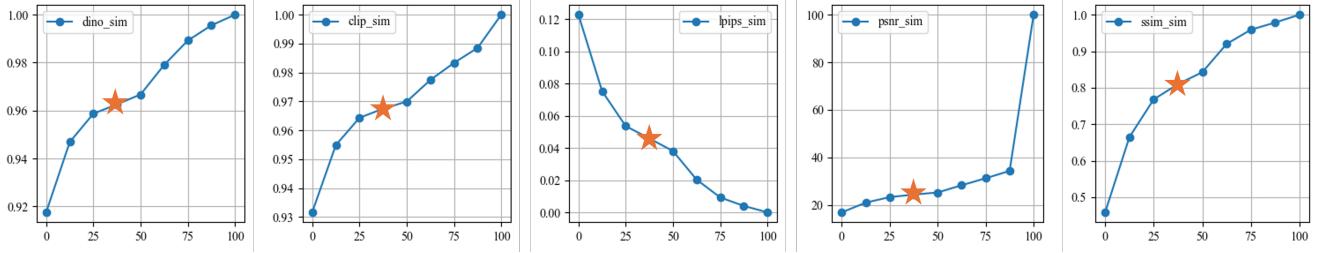
Figure A3. **LTX-Video: Experiments for choosing late stage boundary.** The star sign represents our choice of the late stage boundary. The star sign represents the large model reintroduction points (% from end). We measure Subject Consistency, Background Consistency, Temporal Flickering, Motion Smoothness, Dynamic Degree, Aesthetic Quality, Image Quality, FID, and FVD.

Figure A4. **WAN 2.1: Experiments for choosing late stage boundary.** The star sign represents our choice of the late stage boundary. The star sign represents the large model reintroduction points (% from end). We measure Subject Consistency, Background Consistency, Temporal Flickering, Motion Smoothness, Dynamic Degree, Aesthetic Quality, Image Quality, FID, and FVD.

(LSS) corresponds to not reintroducing the large model. When the reintroduction point is set to 20%, the model preserves fine-grained textures without introducing noticeable artifacts. However, pushing the boundary further to 0% (LSS) introduces visible artifacts-most prominently, a severe artifact on the dog's eye in the last frame. Therefore, reintroducing the large model at 20%, our final choice, offers the optimal balance between video quality and efficiency.

C. U-shaped divergence curve

C.1. Experimental details

We measured the divergence between the velocity fields of the small and large models across timesteps.

The details are as follows. LTX-Video [11] is an image-to-video (I2V) model, and Wan2.1 (1.3B/13B) [42] is a text-to-

Figure A5. **Velocity divergence across diffusion steps.** We measure the divergence between the velocity predictions of the small and large models at every denoising step for LTX-Video and WAN2.1. (a) uses the large model as the main stream, while (b) uses the small model as the main stream. For each timestep, we compare the velocity predicted with the conditional input (cond) and the null input (uncond), corresponding to the CFG-conditioned and unconditioned branches, respectively. Divergence is lowest in the intermediate stage, indicating that the small model denoises reliably there, whereas the early and late stages show higher divergence due to structure formation and detail refinement.

video (T2V) model. We measured 355 videos using the text and image prompt from VBench dataset [17]. We use the default sampler and inference timesteps, following the official repository of each video model. We measured the runtime of LTX-video and WAN2.1 on a NVIDIA A6000 GPU and on a NVIDIA A100 GPU, respectively. In the Figure A6, we provide the illustration of how we measure the divergence between the velocity fields of the small and large models across timesteps. As shown in the figure, we set the sampling trajectory of the large models as a *main stream*, and we fed the previous large model outputs to the large model and the small model during inference. We measure the divergence between the velocity predictions of the small and large models at each time step. We use cosine distance and L2 distance as measurements.

In the Figure A5 (a), we provide the experiment results. Divergence is lowest in the intermediate stage, indicating that the small model can reliably denoise there. Early stage exhibits high variance (structure formation), while late stage shows high mean divergence (detail refinement). This supports allocating the large model to the early and late stages. Interestingly, we found a similar result with the small model as a mainstream.

D. Comprehensive schedule sweep

We further provide the evaluations using FID, FVD, and VBench across various combinations of the small and large models. In Figure A7, we evaluate LTX-Video [11] on the PV dataset [3] and sort all combinations in ascending order of their FID scores. We observe that our LSL strategy consistently outperforms the other combinations. In Figure A9, we present the results on the VBench dataset using the same combination order obtained from the PV dataset.

Likewise, we provide the evaluation results with WAN2.1 [42]. In Figure A9, we evaluate Wan2.1 on the PV dataset and sort all combinations in ascending order of their FID scores. We also observe that our LSL strategy consistently outperforms

Figure A6. **Illustration of velocity divergence across diffusion steps.** We follow the sampling trajectory of the main-stream model (blue) and feed its intermediate outputs into both the large and small models at each timestep. The divergence between their velocity predictions is then computed using cosine distance and L2 distance. This procedure is applied identically to LTX-Video and WAN2.1, using the official inference settings of each model.

the other combinations. In Figure A10, we present the results on the VBench dataset using the same combination order obtained from the PV dataset.

E. Qualitative results

We include the videos corresponding to all figures presented in the main paper in the supplementary material. Please refer to our project page (https://jibin86.github.io/flowblending_project_page/)

F. More Related Work

Beyond the methods introduced in the main paper, several additional approaches have been proposed to accelerate diffusion sampling. Adversarial distillation has been explored as an effective way to reduce the number of inference steps [36, 37, 46]. Based on a deeper understanding of diffusion dynamics, training-free acceleration techniques have also been developed, leveraging step-wise feature reuse [27, 30, 51].

Pan et al. [33] is the most closely related to our work, as they also adopt different models for different denoising stages. However, unlike Pan et al. [33], whose aim is primarily to improve image quality, our goal is to preserve the fidelity of a large video diffusion model while reducing computational cost. In video generation, early-stage misalignment in motion cannot be corrected by a small model in later stages. To tackle this challenge, we analyze the stage-specific behavior of video diffusion models and introduce a stage-aware strategy that determines stage boundaries using capacity-sensitive metrics.

LTX-Video

FID	FVD	SC	BC	TF	MS	DD	AQ	IQ
5.70	752.07	90.79	93.96	95.20	97.74	33.98	44.52	46.14
5.72	818.83	91.06	93.86	94.81	97.55	35.21	45.47	48.00
5.72	764.43	90.90	93.91	95.21	97.78	33.63	44.83	46.62
5.73	809.61	91.04	93.99	94.90	97.61	34.68	45.26	47.62
5.73	834.26	90.96	93.79	94.81	97.56	35.21	45.43	48.12
5.73	813.19	90.81	93.94	95.15	97.76	34.51	44.64	46.24
5.73	773.73	90.94	93.93	95.13	97.70	34.33	44.87	46.64
5.73	835.04	91.13	93.95	94.92	97.64	35.56	45.49	47.87
5.74	772.28	90.85	93.91	95.12	97.67	33.80	44.45	46.24
5.74	788.74	90.93	93.93	95.08	97.76	34.68	45.17	47.41
5.74	801.22	91.10	93.97	94.99	97.64	35.21	44.94	46.92
5.75	801.31	90.95	94.04	94.93	97.53	34.33	44.71	46.50
5.75	759.39	90.66	93.89	95.14	97.63	34.15	44.25	45.77
5.75	769.47	90.71	93.86	95.06	97.56	33.45	44.22	45.83
5.75	782.32	90.90	93.95	95.03	97.76	35.56	45.03	46.75
5.75	825.55	90.67	93.90	95.15	97.73	33.98	44.22	45.84
5.75	828.27	91.02	93.95	94.83	97.56	35.04	45.43	48.06
5.75	812.81	91.15	94.00	94.98	97.66	34.68	45.22	47.40
5.76	822.85	90.92	93.90	94.85	97.54	34.68	45.24	47.63
5.76	764.00	90.93	93.85	95.01	97.72	34.51	45.10	47.33
5.76	853.27	90.51	93.60	94.78	97.59	36.97	44.98	47.07
5.77	809.53	90.97	93.94	94.86	97.52	35.04	44.98	47.17
5.77	825.70	90.19	93.72	94.93	97.57	35.74	43.98	45.90
5.78	838.85	90.72	93.69	94.62	97.45	35.92	45.41	47.80
5.79	827.03	90.30	93.70	94.92	97.60	35.92	44.46	46.21
5.79	840.10	90.86	93.70	94.70	97.49	35.56	45.44	47.99
5.79	828.16	90.52	93.89	95.09	97.61	33.80	44.02	45.43
5.79	829.45	90.06	93.75	94.86	97.45	35.21	43.78	45.47
5.80	838.92	90.82	93.82	94.68	97.45	35.04	45.41	48.11
5.88	901.37	89.63	93.10	93.57	96.87	39.08	45.35	48.25
6.01	874.96	88.46	92.75	93.82	96.93	38.91	43.69	45.52
6.03	915.19	88.60	92.45	92.90	96.63	40.32	45.44	48.14
6.06	970.07	88.02	92.24	92.60	96.43	41.73	45.24	48.03
6.07	904.85	87.95	92.37	93.25	96.65	39.26	43.91	45.30
6.07	955.37	87.95	92.32	92.66	96.36	41.37	44.74	47.09
6.11	943.44	87.56	92.17	93.08	96.50	40.49	43.68	45.17
6.15	963.15	87.45	92.15	92.54	96.24	41.73	44.11	46.00
6.15	1004.22	87.59	92.01	92.40	96.36	42.08	45.04	47.75
6.16	1007.87	87.49	92.05	92.46	96.28	41.55	44.46	46.87
6.21	959.41	87.06	92.05	92.73	96.34	41.37	43.66	45.00
6.22	980.41	86.89	91.71	92.40	96.28	41.55	44.42	46.41
6.22	953.60	86.42	91.75	92.57	96.20	40.67	43.46	44.92
6.24	969.52	86.61	91.83	92.57	96.27	41.02	44.03	45.38
6.25	960.97	86.46	91.72	92.61	96.20	40.49	43.73	44.87
6.26	990.61	86.71	91.78	92.44	96.15	41.90	43.80	45.44
6.28	951.79	86.23	91.70	92.47	96.02	40.67	43.14	44.48

Step (0–39)

Figure A7. **LTX-Video: Comprehensive schedule sweep on the PV dataset.** We evaluate all combinations of small–large scheduling for LTX-Video on the PV dataset and sort the results in ascending order of FID. Our LSL configuration consistently ranks near the top, demonstrating strong performance across both FID and FVD. Other metrics also exhibit a similar overall trend.

7

	SC	BC	TF	MS	DD	AQ	IQ
	97.19 /	96.83 /	96.53 /	98.90 /	18.50 /	60.01 /	70.63 /
	97.01 /	96.43 /	96.50 /	98.87 /	19.51 /	59.87 /	70.74 /
	97.33 /	96.60 /	96.66 /	98.95 /	18.29 /	60.03 /	70.57 /
	97.34 /	96.69 /	96.66 /	98.95 /	18.09 /	60.06 /	70.62 /
	97.15 /	96.39 /	96.54 /	98.90 /	18.09 /	59.88 /	70.62 /
	97.15 /	96.66 /	96.52 /	98.88 /	18.29 /	59.88 /	70.67 /
	97.16 /	96.47 /	96.57 /	98.90 /	18.29 /	59.73 /	70.46 /
	96.90 /	96.37 /	96.44 /	98.82 /	19.31 /	59.57 /	70.50 /
	97.04 /	96.24 /	96.47 /	98.86 /	17.89 /	59.57 /	70.41 /
	97.25 /	96.56 /	96.63 /	98.95 /	18.70 /	60.02 /	70.62 /
	97.25 /	96.58 /	96.63 /	98.94 /	17.89 /	60.07 /	70.64 /
	97.32 /	96.70 /	96.69 /	98.94 /	17.89 /	59.95 /	70.44 /
	97.25 /	96.54 /	96.59 /	98.92 /	17.68 /	59.88 /	70.61 /
	97.30 /	96.49 /	96.65 /	98.93 /	17.89 /	59.74 /	70.39 /
	96.87 /	96.61 /	96.30 /	98.75 /	19.51 /	59.59 /	70.50 /
	96.44 /	96.20 /	95.89 /	98.62 /	23.17 /	59.19 /	70.45 /
	97.14 /	96.32 /	96.21 /	98.81 /	21.34 /	59.84 /	70.63 /
	95.85 /	94.65 /	93.98 /	98.19 /	27.85 /	59.25 /	70.25 /
	95.39 /	94.17 /	93.62 /	98.05 /	31.50 /	58.67 /	69.99 /
	95.38 /	94.33 /	93.71 /	97.99 /	29.07 /	58.96 /	70.17 /
	95.41 /	94.39 /	93.65 /	97.90 /	30.89 /	59.04 /	70.33 /
	95.07 /	93.86 /	93.37 /	97.88 /	31.71 /	58.40 /	70.03 /
	95.36 /	94.37 /	93.62 /	97.85 /	30.49 /	58.72 /	70.02 /
	94.99 /	93.86 /	93.32 /	97.87 /	31.71 /	58.55 /	69.95 /
	95.16 /	94.12 /	93.47 /	97.89 /	31.50 /	58.68 /	70.03 /
	95.35 /	94.28 /	93.62 /	97.95 /	31.10 /	59.08 /	70.37 /
	95.32 /	94.13 /	93.61 /	97.92 /	31.30 /	58.73 /	70.03 /
	95.12 /	93.93 /	93.35 /	97.82 /	33.13 /	58.60 /	70.07 /
	95.22 /	94.33 /	93.50 /	97.92 /	32.11 /	59.02 /	70.36 /
	95.06 /	94.10 /	93.37 /	97.87 /	32.11 /	58.93 /	70.26 /
	95.13 /	94.15 /	93.43 /	97.89 /	31.71 /	59.08 /	70.35 /
	95.13 /	94.19 /	93.49 /	97.87 /	31.91 /	58.85 /	70.13 /
	95.19 /	94.29 /	93.49 /	97.89 /	32.11 /	58.84 /	70.10 /
	94.99 /	93.87 /	93.33 /	97.83 /	32.11 /	58.60 /	70.04 /

Figure A8. **LTX-Video: VBench evaluation under the same schedule ordering.** Using the ordering derived from the PV dataset sweep, we evaluate LTX-Video on VBench metrics. The relative ranking remains consistent, and the LSL configuration again shows robust performance across diverse quality dimensions. Other metrics also exhibit a similar overall trend.

Figure A9. **WAN 2.1: Comprehensive schedule sweep on the PV dataset.** We apply the same exhaustive combination sweep to WAN 2.1 and sort all configurations by FID. Similar to LTX-Video, the LSL strategy consistently outperforms other schedules across both FID. Other metrics also exhibit a similar overall trend.

FID	SC	BC	TF	MS	DD	AQ	IQ
7.48 /	93.68 /	94.98 /	91.37 /	94.39 /	38.03 /	57.42 /	65.13 /
7.49 /	93.59 /	94.77 /	91.35 /	94.43 /	39.08 /	57.45 /	65.75 /
7.49 /	93.62 /	94.89 /	91.31 /	94.32 /	38.73 /	57.40 /	64.55 /
7.50 /	93.46 /	94.95 /	91.34 /	94.40 /	39.08 /	57.41 /	65.56 /
7.51 /	93.73 /	94.91 /	91.40 /	94.42 /	39.08 /	57.42 /	65.23 /
7.51 /	93.52 /	94.74 /	91.35 /	94.32 /	38.38 /	57.27 /	64.91 /
7.52 /	93.44 /	94.81 /	91.33 /	94.33 /	39.44 /	57.25 /	65.37 /
7.52 /	93.34 /	94.37 /	90.84 /	94.04 /	37.68 /	56.84 /	67.56 /
7.54 /	93.74 /	95.03 /	91.35 /	94.41 /	39.08 /	57.54 /	64.83 /
7.54 /	93.64 /	94.88 /	91.28 /	94.31 /	39.44 /	57.42 /	64.60 /
7.54 /	93.54 /	94.79 /	91.37 /	94.42 /	38.73 /	57.47 /	65.72 /
7.54 /	93.19 /	94.05 /	90.77 /	93.85 /	37.32 /	56.20 /	67.22 /
7.55 /	93.68 /	94.87 /	91.23 /	94.34 /	38.73 /	57.56 /	64.93 /
7.55 /	93.15 /	94.19 /	90.75 /	93.76 /	36.97 /	56.21 /	66.96 /
7.55 /	93.09 /	94.23 /	90.74 /	93.88 /	38.38 /	56.41 /	67.65 /
7.57 /	93.31 /	94.33 /	90.76 /	93.88 /	38.03 /	56.26 /	67.36 /

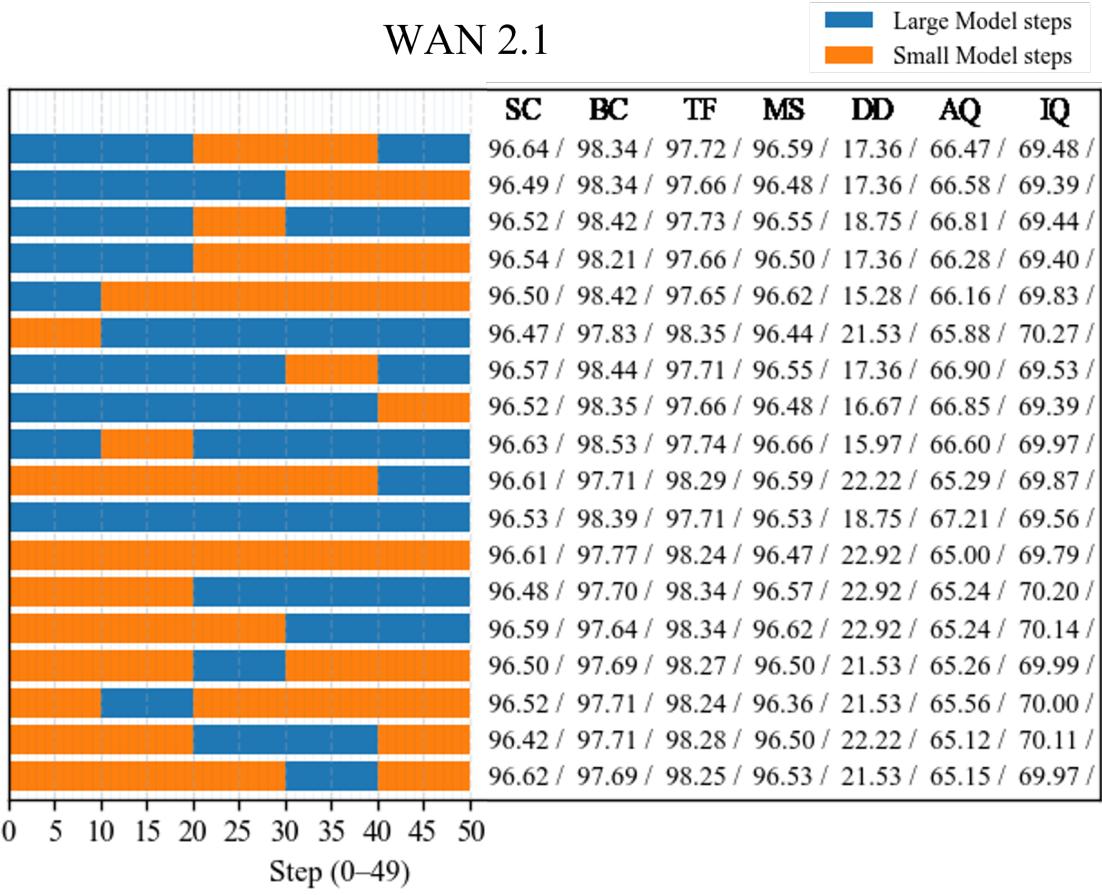|	SC	BC	TF	MS	DD	AQ	IQ
	96.64 /	98.34 /	97.72 /	96.59 /	17.36 /	66.47 /	69.48 /
	96.49 /	98.34 /	97.66 /	96.48 /	17.36 /	66.58 /	69.39 /
	96.52 /	98.42 /	97.73 /	96.55 /	18.75 /	66.81 /	69.44 /
	96.54 /	98.21 /	97.66 /	96.50 /	17.36 /	66.28 /	69.40 /
	96.50 /	98.42 /	97.65 /	96.62 /	15.28 /	66.16 /	69.83 /
	96.47 /	97.83 /	98.35 /	96.44 /	21.53 /	65.88 /	70.27 /
	96.57 /	98.44 /	97.71 /	96.55 /	17.36 /	66.90 /	69.53 /
	96.52 /	98.35 /	97.66 /	96.48 /	16.67 /	66.85 /	69.39 /
	96.63 /	98.53 /	97.74 /	96.66 /	15.97 /	66.60 /	69.97 /
	96.61 /	97.71 /	98.29 /	96.59 /	22.22 /	65.29 /	69.87 /
	96.53 /	98.39 /	97.71 /	96.53 /	18.75 /	67.21 /	69.56 /
	96.61 /	97.77 /	98.24 /	96.47 /	22.92 /	65.00 /	69.79 /
	96.48 /	97.70 /	98.34 /	96.57 /	22.92 /	65.24 /	70.20 /
	96.59 /	97.64 /	98.34 /	96.62 /	22.92 /	65.24 /	70.14 /
	96.50 /	97.69 /	98.27 /	96.50 /	21.53 /	65.26 /	69.99 /
	96.52 /	97.71 /	98.24 /	96.36 /	21.53 /	65.56 /	70.00 /
	96.42 /	97.71 /	98.28 /	96.50 /	22.22 /	65.12 /	70.11 /
	96.62 /	97.69 /	98.25 /	96.53 /	21.53 /	65.15 /	69.97 /

Figure A10. **WAN 2.1: VBench evaluation under the same schedule ordering.** Using the schedule ordering obtained from the PV dataset, we report WAN 2.1 performance on VBench. The LSL schedule again provides strong performance across multiple VBench dimensions, confirming its general effectiveness.