ECOVIDEO++: CHARACTERIZING AND EXPLOITING TRI-DIMENSIONAL CAPABILITY BOUNDARIES FOR COLLABORATIVE VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Transformers (DiTs) achieve strong video generation quality but incur substantial inference costs due to repeated processing of dense spatiotemporal tokens. Existing large–small model collaboration methods reduce this cost by assigning different denoising stages to models of different capacities, while still treating the entire video uniformly at each step. We observe that the capability difference between large and small DiTs is highly non-uniform not only along the denoising trajectory, but also across temporal segments and spatial regions. Based on this observation, we propose EcoVideo++, a training-free hierarchical capacity orchestration framework that progressively allocates large-model computation at step, temporal, and spatial levels. The small model provides global predictions, while the large model selectively refines difficult regions with preserved spatiotemporal context and smooth prediction fusion. Experiments on Wan2.1, Wan2.2, and CogVideoX demonstrate that EcoVideo++ achieves 2.29–2.93× speedup while keeping VBench within 0.0027 of the corresponding large-only models.

1 INTRODUCTION

Video diffusion models (VDMs) have advanced video interaction and world modeling, demonstrating strong spatiotemporal modeling capabilities (Ho et al., 2022; Wan et al., 2025; Yang et al., 2025b; NVIDIA et al., 2025). However, their substantial computational demands lead to high inference latency (Zhang et al., 2025). Improving inference efficiency is therefore important for their broader deployment.

To improve VDM inference efficiency, prior work has explored strategies such as caching (Lv et al., 2025; Liu et al., 2025; Zhao et al., 2025), distillation (Zhai et al., 2024; Yin et al., 2025), and sparse attention (Xi et al., 2025; Zhang et al., 2025). A newer approach, Collaborative Video Diffusion (Cheng et al., 2025; Xie et al., 2025), assigns denoising steps to models with different parameter counts from the same family that share a VAE, such as Wan2.1-14B and Wan2.1-1.3B (Wan et al., 2025). The larger model has greater modeling capacity and can handle fine texture details, but incurs higher computational cost. In contrast, the smaller model has fewer parameters and is limited to generating simpler structures.

However, these methods coordinate models through a single-dimensional switching strategy (Pan et al., 2025; Cheng et al., 2025), leaving a central question unanswered: ***where do the capability boundaries of large and small models lie?***

In this paper, we systematically analyze the capability boundaries of large and small models to guide efficient collaboration in video generation. Specifically, we use the normalized difference between their denoising predictions under the same generation state to examine three perspectives:

(1) **Semantics Depth**: Global semantic structure forms jointly across space and time during early high-noise denoising; tracking the model gap at the whole-video level shows that it decreases over denoising but remains high longer for complex content.

(2) **Temporal Coverage**: Since video generation jointly models multiple frames within a single video sequence, we compare the model gap across temporal segments at the same denoising step and find that it is larger in some segments than in others.

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.

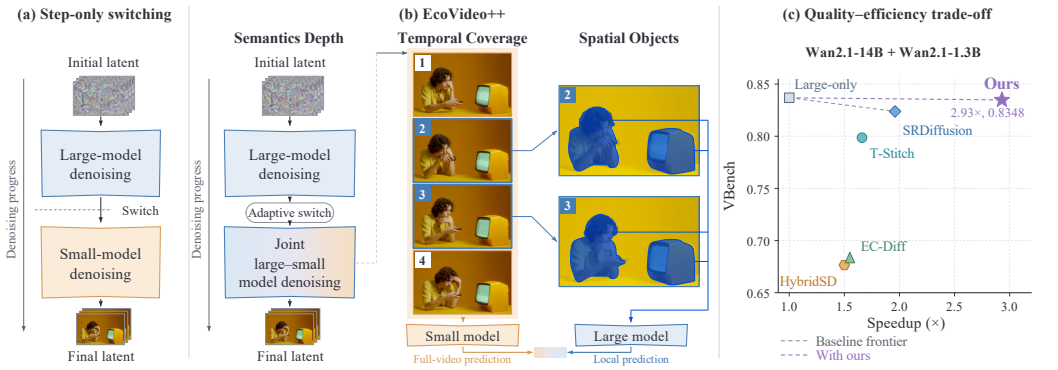


Figure 1: (a) Single-dimensional model switching. (b) EcoVideo++'s tri-dimensional collaboration space. (c) Quality–efficiency trade-offs across methods.

(3) **Spatial Objects**: Within each generated frame, objects and local details vary in complexity, and comparing the model gap across spatial regions shows that larger gaps concentrate around intricate objects rather than being evenly distributed.

Guided by the tri-dimensional analysis, we propose **EcoVideo++**, a training-free framework that allocates large-model computation across denoising stages, temporal segments, and spatial regions. It first determines when to transition from full-video large-model denoising to joint large–small denoising, then identifies the temporal segments and local object regions that require large-model predictions. During joint denoising, the small model predicts the video, while the large model processes selected regions with preserved spatiotemporal context. EcoVideo++ smoothly fuses their predictions to reduce large-model computation while maintaining global video consistency.

Compared with single-dimensional switching method, EcoVideo++ uses the tri-dimensional capability analysis to determine both when the large model participates and which temporal segments and spatial object regions it processes. Across three model pairs spanning Wan2.1, Wan2.2, and CogVideoX, it achieves 2.29–2.93× end-to-end speedups over large-only inference, with VBench scores no more than 0.0027 below their large-only counterparts.

Our main contributions are as follows:

- We systematically characterize the capability boundaries between large and small VDMs from three perspectives: **Semantics Depth**, **Temporal Coverage**, and **Spatial Objects**.
- Guided by these observations, we propose **EcoVideo++**, a training-free tri-dimensional collaborative video diffusion framework that selectively allocates large-model computation across denoising stages, temporal segments, and spatial regions, while preserving spatiotemporal context and smoothly fusing large–small predictions.
- We evaluate EcoVideo++ on three model pairs spanning Wan2.1, Wan2.2, and CogVideoX. EcoVideo++ achieves **2.29–2.93**× end-to-end speedups over large-only inference, with VBench degradation of at most **0.0027**.

2 PRELIMINARY

Video Diffusion Modeling. Video diffusion models generate videos by progressively denoising a VAE latent $z_i \in \mathbb{R}^{C\times F\times H\times W}$, where C is the channel count, F is the number of latent frames, and H, W are the spatial dimensions (Wan et al., 2025; Yang et al., 2025b). The step index i tracks the evolution of the entire video from high noise toward finer detail. Within each state, f indexes a temporal frame, while p indexes one of the HW spatial tokens in that frame; we write the corresponding token as $z_{i,f,p} \in \mathbb{R}^C$. The indices i, f, and p therefore locate generation at the global, temporal, and spatial scales, respectively.

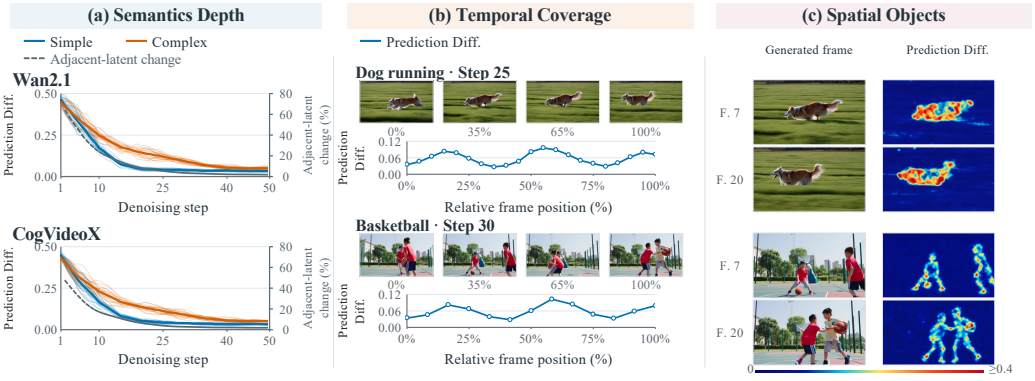


Figure 2: Tri-dimensional analysis of large–small model capability boundaries.

To update this structured latent, a video DiT M (Peebles & Xie, 2023) processes the full state z_i under condition c and predicts an output for every frame and token:

$$u_i^{M,c} = \mathcal{D}_M(z_i, i, c) = \left(u_{i,f,p}^{M,c}\right)_{1\le f\le F,\ 1\le p\le HW}, \tag{1}$$

where $\mathcal{D}_M$ denotes the denoising network. Because each output is predicted from the full latent state, it can incorporate context across frames and spatial positions. Classifier-free guidance (CFG) then combines the conditional and unconditional predictions into $\widetilde{u}_i^M$ (Ho & Salimans, 2022), which the scheduler uses to obtain z_{i+1}.

Normalized Prediction Difference. A large video DiT L and a small video DiT S from the same model family share a VAE and therefore operate in the same latent space (Cheng et al., 2025). Given the same state z_i, condition c, and CFG settings, they produce $\widetilde{u}_i^L$ and $\widetilde{u}_i^S$ at corresponding frame-token positions. Small differences between these predictions indicate similar denoising behavior, whereas larger differences point to potential capability boundaries between the two models. To quantify these differences at different granularities, we use

$$D_i(\mathcal{A}) = \frac{\mathrm{mean}_{\mathcal{A}}\left|\widetilde{u}_i^L - \widetilde{u}_i^S\right|}{\mathrm{mean}_{\Omega}\left|\widetilde{u}_i^L\right| + \varepsilon}, \tag{2}$$

where Ω contains all frame-token positions and $\mathcal{A} \subseteq \Omega$ specifies the positions of interest. Both means also average over channels, and ε ensures numerical stability. Normalizing by L's full-video prediction provides a consistent reference for comparing the models at step i. Varying $\mathcal{A}$ enables analysis across three dimensions: the full video across denoising steps for semantics depth, individual frames for temporal coverage, and spatial regions within frames for objects.

3 Tri-Dimensional Capability Boundary Analysis and Modeling

We use the *Normalized Prediction Difference* (Eq. 2) as a unified indicator of capability differences between large and small models across three dimensions. The resulting analysis reveals where their capability boundaries lie and guides the allocation of large-model computation. The analysis protocol is detailed in Appendix B.1.

Semantics Depth Analysis. Step-wise collaboration methods such as T-Stitch (Pan et al., 2025) and SRDiffusion (Cheng et al., 2025) assign one model to the entire sample at each denoising step. Their switching policies do not directly measure how the capability gap between the large and small models varies with the semantic complexity of the current content. To examine this gap, we evaluate both models at the same state z_i and compute the normalized prediction difference over the full video:

$$D_i(\Omega) = \frac{\mathrm{mean}_{\Omega}\left|\widetilde{u}_i^L - \widetilde{u}_i^S\right|}{\mathrm{mean}_{\Omega}\left|\widetilde{u}_i^L\right| + \varepsilon}. \tag{3}$$

Because global semantic structure is formed across frames and spatial positions, averaging over Ω captures the video-wide prediction difference at step i. We compute this quantity along the large-model denoising trajectory, using the same configuration for both models.

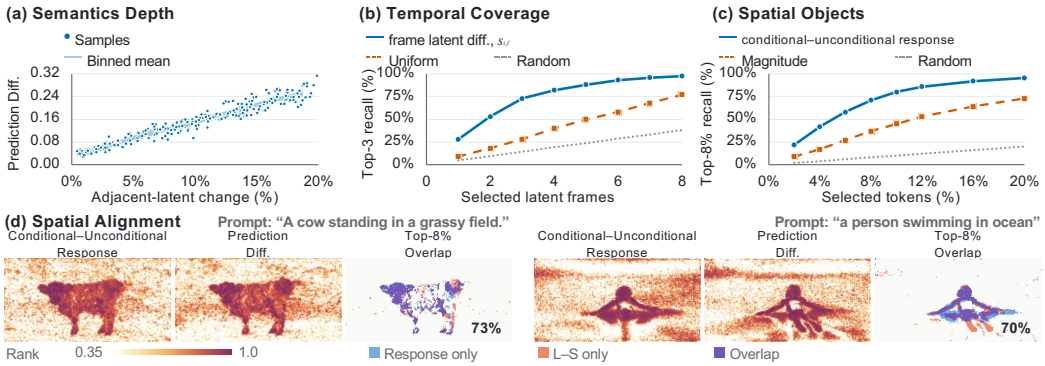


Figure 3: Tri-dimensional metrics and validation of large–small model capability boundaries.

Figure 2(a) shows that $D_i(\Omega)$ is larger in early denoising and decreases in later steps for both Wan2.1 and CogVideoX. We further group prompts with a single subject and simple actions as Simple, and prompts with multiple interacting subjects or complex actions as Complex. The difference becomes small earlier for Simple content, whereas it remains high for more steps for Complex content.

The capability boundary along Semantics Depth is therefore content-dependent: the stage at which the small model's full-video prediction approaches the large model's varies with the video.

To model this content-dependent boundary during inference, we track how the large model's estimate of the video latent changes between denoising steps. Let $\widehat{z}_i^L$ denote the estimate derived from L's prediction at step i. Unlike changes in the noisy state z_i, which also reflect the noise schedule, changes between consecutive estimates indicate how much the generated content is still being revised. We define the normalized adjacent-latent change as

$$r_i = \frac{\mathrm{RMS}\big(\widehat{z}_i^L - \widehat{z}_{i-1}^L\big)}{\mathrm{RMS}\big(\widehat{z}_{i-1}^L\big) + \varepsilon}. \tag{4}$$

A larger r_i indicates that global semantic content is still undergoing substantial revision. As shown in Figure 3(a), larger adjacent-latent changes tend to accompany larger whole-video normalized prediction differences $D_i(\Omega)$. This relationship supports using r_i to identify when the content estimate has stabilized enough to transition to joint denoising.

Temporal Coverage Analysis. Once the transition to joint denoising is determined, we ask which temporal segments still benefit from large-model predictions. The full-video difference $D_i(\Omega)$ tracks the overall gap between the models, but its average over frames cannot locate that gap in time. To examine this variation at step i, let $\Omega_f = \{(f,p) \mid 1 \le p \le HW\}$ denote all positions in frame f. The frame-wise normalized prediction difference is

$$D_i(\Omega_f) = \frac{\mathrm{mean}_{\Omega_f} \left|\widetilde{u}_i^L - \widetilde{u}_i^S\right|}{\mathrm{mean}_{\Omega} \left|\widetilde{u}_i^L\right| + \varepsilon}. \tag{5}$$

Because the frames share the same text condition, changes in $D_i(\Omega_f)$ reveal how the prediction gap is distributed over time. Figure 2(b) shows that, in the running-dog and basketball scenes, segments with larger differences coexist with segments where the models' predictions are closer.

The capability boundary along Temporal Coverage is therefore segment-dependent: at the same step, different parts of a video exhibit different large–small model gaps.

This observation motivates selecting temporal segments for complementary large-model predictions. Computing $D_i(\Omega_f)$ during inference, however, would require evaluating the large model on every frame. We therefore use the neighboring-frame latent difference as an online selection signal:

$$s_{i,f} = \frac{1}{|\mathcal{N}(f)|} \sum_{f' \in \mathcal{N}(f)} \mathrm{mean}_{p,\mathrm{ch}} \left|z_{i,f,p} - z_{i,f',p}\right|, \tag{6}$$

where $\mathcal{N}(f)$ contains the valid adjacent frames of frame f. This signal measures local variation within the current video latent z_i. As shown in Figure 3(b), ranking frames by $s_{i,f}$ recalls frames

with large prediction differences more effectively than uniform or random selection. We therefore prioritize frames with higher scores and merge adjacent selections into temporal segments for complementary large-model predictions.

Spatial Objects Analysis. Even within a selected frame, averaging over spatial positions can conceal differences around individual objects. We therefore evaluate the normalized prediction difference at each position (f, p):

$$D_i\big(\{(f,p)\}\big) = \frac{\mathrm{mean}_{\mathrm{ch}} \left| \widetilde{u}^{L}_{i,f,p} - \widetilde{u}^{S}_{i,f,p} \right|}{\mathrm{mean}_{\Omega} \left| \widetilde{u}^{L}_{i} \right| + \varepsilon}. \tag{7}$$

Figure 2(c) shows larger differences around the dog's body in the running-dog scene and around people in the basketball scene, while the surrounding backgrounds show smaller differences.

The capability boundary along Spatial Objects is therefore object-dependent: subjects and local details retain a larger large–small model gap than other regions of the same frame.

Since larger prediction differences concentrate around objects, we use the small model's CFG predictions (Ho & Salimans, 2022) to locate text-related regions without evaluating the large model at every position. The conditional and unconditional predictions share the same latent state, so their difference reflects the local effect of the text condition. For a selected temporal segment I, we define

$$G_j(f,p) = \mathrm{mean}_{\mathrm{ch}} \left| u^{S,c}_{j,f,p} - u^{S,\varnothing}_{j,f,p} \right|, \overline{G}_{j,I}(p) \quad = \frac{1}{|I|} \sum_{f \in I} G_j(f,p). \tag{8}$$

The aggregated map $\overline{G}_{j,I}$ highlights positions responsive to the prompt across the selected frames. We use current predictions at the first joint step ($j = i$) and the cached map thereafter ($j = i - 1$). Figure 3(c–d) shows that this signal recalls more positions with large prediction differences than the baselines and overlaps spatially with the large–small prediction gap, supporting its use for selecting local large-model regions.

4 Tri-dimensional collaborative video diffusion framework

4.1 Overview

The capability analysis identifies where the Normalized Prediction Difference between the large and small models remains high. Evaluating this difference during inference, however, would require full-video predictions from both models at each step. EcoVideo++ therefore uses the online signals established in that analysis to make three nested decisions: *Semantics Depth* determines when to leave full-video large-model denoising, *Temporal Coverage* selects the segments that receive large-model predictions, and *Spatial Objects* locates regions within those segments. As shown in Figure 4, the large model initially processes the latent. After the transition, the small model predicts the full video, while the large model processes selected regions with surrounding spatiotemporal context. Their predictions are fused before a scheduler update. The framework requires no retraining.

4.2 Semantic-aware Adaptive Transition

The Semantics Depth analysis shows that the whole-video model gap decreases at a content-dependent rate. A fixed switching step cannot distinguish a video whose global structure has stabilized from one whose content is still changing. We therefore use the normalized adjacent-latent change r_i defined in the capability analysis. It compares the large model's estimates $\widehat{z}^{L}_{i}$ and $\widehat{z}^{L}_{i-1}$ at consecutive denoising steps; each is converted from the model prediction to an estimate of the clean latent. Thus, r_i tracks revisions to the estimated final video rather than changes between the noisy states z_i and z_{i-1}, which also reflect the noise schedule.

The estimate from the first full-video large-model step serves as the reference; r_i is first computed after the next large-model prediction. At subsequent full-video large-model steps, a value $r_i \leq \tau$ increments a stability counter, whereas a larger value resets it. We switch only after P consecutive stable values, so a brief plateau does not trigger joint denoising. The following step is i_{sw}: its input is the latent produced by the preceding scheduler update, and the trajectory continues without

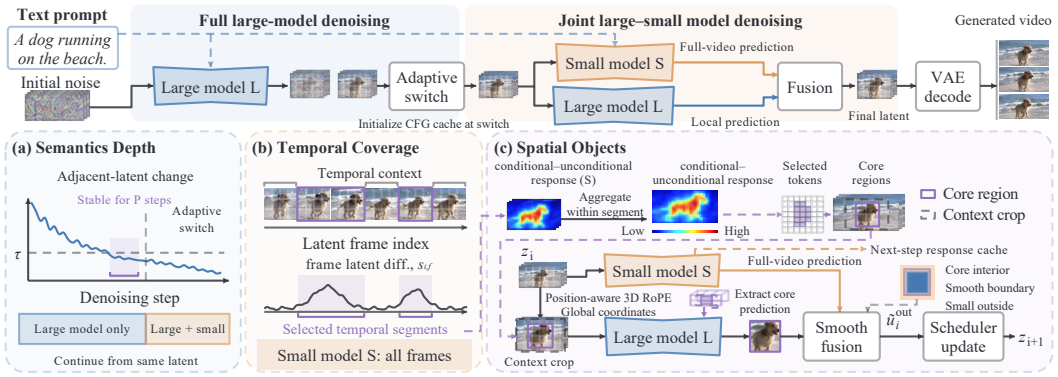


Figure 4: Tri-dimensional collaborative video diffusion framework, EcoVideo++.

restarting. From this step onward, the small model supplies a full-video prediction and the large model is reserved for selected spatiotemporal regions. If the stability condition is never met, full-video large-model denoising continues. Thus, the stage decision adapts to the current sample while preserving large-model coverage during substantial video-wide revision.

4.3 Temporal Segment Selection

After the stage transition, the Temporal Coverage analysis leaves a second question: which parts of the video benefit from large-model computation at the current step? The frame-wise Normalized Prediction Difference would answer this, but computing it online would require a full-video large-model pass. Instead, we evaluate the neighboring-frame latent difference $s_{i,f}$, defined in the analysis, on the current z_i. The score averages the change over spatial positions and channels; an interior frame uses both adjacent frames, while a boundary frame uses its available neighbor. It therefore highlights content transitions without additional model evaluations. The analysis shows that ranking by this signal recalls frames with larger model gaps better than uniform or random.

Given frame budget ρ_f, we select the highest-scoring $\lceil \rho_f F \rceil$ latent frames. We then merge adjacent selections into segments, allowing each local large-model pass to model a temporal range rather than isolated frames. If the merged segments exceed the permitted number of local predictions, we rank them by their mean frame scores and retain the highest-ranked ones. Their set $\mathcal{I}_i$ determines where spatial selection is performed at this step. The frame budget controls the initial selection; the actual large-model input also includes the context added after spatial cores are constructed.

4.4 Spatial Region Selection

The Spatial Objects analysis shows that larger model gaps concentrate around subjects and local details within a frame. For each selected segment $I \in \mathcal{I}_i$, we use the small model's conditional–unconditional response $\overline{G}_{j,I}$, defined in the analysis, to locate text-responsive positions from predictions already needed for classifier-free guidance (CFG) (Ho & Salimans, 2022). Aggregating the per-frame response over I gives a spatial routing map shared by the segment's frames.

At the first joint step $i = i_{\text{sw}}$, the small model processes the full latent and supplies the current response ($j = i$) for same-step region selection. Thereafter, the preceding step's cached per-frame response ($j = i - 1$) is aggregated over the current segments $\mathcal{I}_i$, while the current small-model response refreshes the cache. Keeping the per-frame map allows temporal and spatial choices to change independently across steps.

For each I, we normalize and smooth $\overline{G}_{j,I}$ on the latent spatial grid, retain the highest-scoring positions under budget ρ_s, and expand their mask to connect nearby responses. We select the connected component with the largest cumulative response; its bounding box and temporal range I define a spatiotemporal *core*. The union of the cores is K_i, where local large-model predictions can be written back. Because ρ_f and ρ_s apply before bounding boxes and context expansion, they guide routing rather than determine the final crop sizes.

4.5 JOINT DENOISING

A core omits information the video DiT may need to predict its content. We therefore expand each core in time and space, clip and align the crop to valid latent and patch boundaries, and retain the original spatiotemporal coordinates for its tokens. The large model processes this context crop under the same denoising step and text condition as the small model. Only its CFG-guided predictions corresponding to the core are extracted; the context participates in computation but is not written back. Meanwhile, the small model supplies a current prediction at every frame-token position.

Finally, we blend each extracted large-model prediction with the small-model prediction. Let $\widetilde{u}^{L,\mathrm{local}}_{i,f,p}$ denote the core prediction and let $\alpha_i(f,p)$ equal one in the core interior and taper toward its temporal and spatial boundaries. The prediction passed to the scheduler is

$$\widetilde{u}^{\mathrm{out}}_{i,f,p} = \begin{cases} \widetilde{u}^{L}_{i,f,p}, & i < i_{\mathrm{sw}}, \\ (1-\alpha_i(f,p))\widetilde{u}^{S}_{i,f,p} + \alpha_i(f,p)\widetilde{u}^{L,\mathrm{local}}_{i,f,p}, & i \geq i_{\mathrm{sw}},\ (f,p) \in K_i, \\ \widetilde{u}^{S}_{i,f,p}, & i \geq i_{\mathrm{sw}},\ (f,p) \notin K_i. \end{cases} \tag{9}$$

The boundary transition avoids an abrupt change between local large-model predictions and the surrounding small-model prediction. The scheduler then advances the latent using $\widetilde{u}^{\mathrm{out}}_i$ to obtain z_{i+1}. Budgets, count limits, crop expansion, and fusion widths are specified in Appendix B.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and baselines. We evaluate EcoVideo++ on three large–small model pairs from Wan2.1 (Wan et al., 2025), Wan2.2 (Wan Team, 2025), and CogVideoX (Yang et al., 2025b). Wan2.1-14B and Wan2.2-T2V-A14B are each paired with Wan2.1-1.3B, and CogVideoX-5B is paired with CogVideoX-2B. We compare with T-Stitch (Pan et al., 2025), HybridSD (Yan et al., 2024), SRDiffusion (Cheng et al., 2025), and EC-Diff (Xie et al., 2025); large-only inference serves as the quality and latency reference. Within each pair, all methods use the same large-model checkpoint and generation settings, and collaborative methods use the same small-model checkpoint.

Evaluation and implementation. We evaluate on the full VBench prompt set (Huang et al., 2024), using five shared seeds per prompt and reporting mean scores. VBench aggregates 16 dimensions into Quality and Semantic scores, weighted by 0.8 and 0.2, respectively. Table 1 also reports six individual dimensions; complete results appear in Appendix C. All experiments use one NVIDIA H200 GPU with batch size 1. End-to-end latency includes prompt encoding, denoising, and VAE decoding, but excludes model loading and disk I/O. Speedup is measured against the corresponding large-only model. Unless stated otherwise, the stage-transition threshold is $\tau = 0.15$; other generation and implementation settings are given in Appendix B.

5.2 MAIN RESULTS

Generation quality and efficiency. Table 1 compares quality and end-to-end latency across three model pairs. EcoVideo++ achieves the highest VBench and lowest latency among accelerated methods on all pairs. It reaches 0.8348 VBench with a $2.93\times$ speedup on Wan2.1, and $2.75\times$ and $2.29\times$ speedups on Wan2.2 and CogVideoX. VBench remains within 0.0027 of large-only inference. Compared with SRDiffusion, EcoVideo++ reduces latency by 27.9%–34.4% while improving VBench and subject/background consistency. These results demonstrate a better quality–efficiency trade-off across VDM families through tri-dimensional large-model allocation.

Qualitative comparison. Figure 5 illustrates how the methods handle local object structure in two scenes. In the trumpet example, T-Stitch blurs or duplicates the hand–instrument contours, and EC-Diff obscures details in deep shadow. EcoVideo++ retains a visible performer and a clearer instrument outline across the sampled frames. In the motorcycle–bus example, its vehicle shapes and wheel details are more recognizable than in the darker EC-Diff output. These examples are consistent with the subject and background consistency results and illustrate the role of complementary local large-model predictions during joint denoising.

Table 1: Generation quality and end-to-end efficiency across three model pairs. Speedup is relative to the corresponding large-only model. Bold marks the best result among accelerated methods.

Method	Subj. Cons. ↑	Back. Cons. ↑	Mot. Smooth. ↑	Aes. Qual. ↑	Img. Qual. ↑	Overall Cons. ↑	VBench ↑	Latency (s) ↓	Speedup ↑
Wan2.1-14B + Wan2.1-1.3B									
Large-only	0.9627	0.9684	0.9752	0.6347	0.7216	0.9218	0.8369	1977.86	1.00×
T-Stitch (2025)	0.9102	0.9138	0.9219	0.5665	0.6631	0.8690	0.7986	1190.28	1.66×
HybridSD (2024)	0.7781	0.7862	0.7826	0.4768	0.5514	0.7393	0.6765	1320.18	1.50×
SRDiffusion (2025)	0.9342	0.9416	0.9491	0.6038	0.6872	0.8875	0.8238	1010.13	1.96×
EC-Diff (2025)	0.7978	0.8114	0.7967	0.4920	0.5679	0.7566	0.6832	1276.04	1.55×
Ours	**0.9589**	**0.9652**	**0.9728**	**0.6527**	**0.7384**	**0.9326**	**0.8348**	**674.33**	**2.93×**
Wan2.2-T2V-A14B + Wan2.1-1.3B									
Large-only	0.9716	0.9741	0.9813	0.6578	0.7348	0.9318	0.8423	1620.18	1.00×
T-Stitch	0.9167	0.9186	0.9269	0.5792	0.6681	0.8834	0.8084	1012.47	1.60×
HybridSD	0.9003	0.8967	0.9091	0.5573	0.6479	0.8615	0.7837	1080.27	1.50×
SRDiffusion	0.9436	0.9481	0.9542	0.6127	0.7025	0.9018	0.8267	900.18	1.80×
EC-Diff	0.9169	0.9188	0.9221	0.5794	0.6733	0.8736	0.7920	1012.54	1.60×
Ours	**0.9689**	**0.9702**	**0.9788**	**0.6715**	**0.7513**	**0.9384**	**0.8396**	**590.13**	**2.75×**
CogVideoX-5B + CogVideoX-2B									
Large-only	0.9254	0.9281	0.9462	0.5937	0.6815	0.8837	0.7784	860.25	1.00×
T-Stitch	0.8631	0.8571	0.8828	0.5165	0.6116	0.8119	0.7479	560.24	1.54×
HybridSD	0.8329	0.8386	0.8420	0.4980	0.5908	0.7917	0.7253	640.19	1.34×
SRDiffusion	0.8792	0.8894	0.9021	0.5486	0.6328	0.8346	0.7708	520.16	1.65×
EC-Diff	0.9072	0.9080	**0.9542**	0.4855	0.5655	0.2281	0.7208	612.24	1.41×
Ours	**0.9218**	**0.9246**	0.9417	**0.6124**	**0.6973**	**0.8972**	**0.7769**	**375.18**	**2.29×**

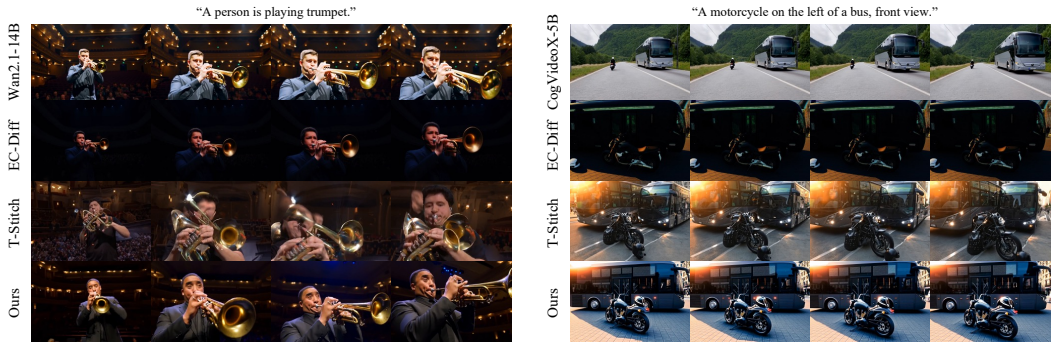


Figure 5: Qualitative comparison on Wan2.1 (left) and CogVideoX (right), with four frames per method. Rows show large-only inference, EC-Diff, T-Stitch, and EcoVideo++, from top to bottom.

5.3 Ablation Study

We ablate EcoVideo++ on Wan2.1-14B paired with Wan2.1-1.3B under the protocol in Section 5.1. Bold marks the best value in each metric column, excluding large-only inference.

Impact of collaboration granularity. We test the three allocation dimensions cumulatively: a denoising-stage decision for Semantics Depth, followed by segment selection for Temporal Coverage and region selection for Spatial Objects. Stage-wise switching alone reaches 0.8147 VBench because it assigns the whole video to one model at each step. Adding temporal selection raises VBench to 0.8235, and adding spatial selection yields 0.8348 with a 2.93× speedup (Table 3). This progression shows that allocating large-model computation within the video improves both VBench and speedup beyond stage-wise switching in this setting.

Table 2: Ablation of the three online allocation metrics.

Allocation	Method	VBench ↑	Quality ↑	Semantic ↑
Overall	Fixed routing	0.7972	0.8173	0.7168
	EcoVideo++	**0.8348**	**0.8551**	**0.7535**
Semantics Depth (r_i)	Fixed switching	0.8108	0.8316	0.7274
	SRDiffusion criterion (2025)	0.8288	0.8492	0.7473
Temporal Coverage ($s_{i,f}$)	Uniform frame selection	0.8235	0.8437	0.7428
	Optical-flow magnitude (2020)	0.8265	0.8464	0.7468
Spatial Objects ($\overline{G}_{j,I}$)	Uniform token selection	0.8102	0.8298	0.7319
	Token attention entropy (2025)	0.8216	0.8418	0.7406

Table 3: Collaboration granularity. Routing dimensions are added cumulatively.

Variant	VBench ↑	Latency(s) ↓	Speedup↑
Large-only	0.8369	1977.86	1.00×
+ Step only	0.8147	1586.82	1.25×
+ Temporal	0.8235	1244.12	1.59×
+ Spatial (full)	**0.8348**	**674.33**	**2.93×**

Table 4: Joint denoising. Components are added cumulatively.

Variant	VBench ↑	Latency(s) ↓
Core crop + hard replace	0.8127	**640.18**
+ Context expansion	0.8216	660.25
+ Position-aware 3D RoPE	0.8294	668.19
+ Smooth fusion (full)	**0.8348**	674.33

Context and fusion in joint denoising. Local large-model predictions must retain surrounding context and align with the small model's full-video prediction. Table 4 compares core cropping with hard replacement against cumulative variants that add context expansion, crop-origin indexing of 3D RoPE (Su et al., 2021), and smooth fusion. Context expansion supplies neighboring tokens, while smooth fusion blends local large-model predictions with the surrounding small-model prediction. The full configuration reaches 0.8348 VBench at 674.33 seconds, compared with 0.8127 at 640.18 seconds for core cropping with hard replacement.

Impact of online allocation signals. To test the signals, we replace one at a time while keeping the other two and matching the routing budgets. We compare the clean-latent change r_i with fixed switching and the SRDiffusion criterion (Cheng et al., 2025); the neighboring-frame latent difference $s_{i,f}$ with uniform selection and optical-flow magnitude (Teed & Deng, 2020); and the conditional–unconditional response $\overline{G}_{j,I}$ with uniform token selection and token attention entropy (Zou et al., 2025). The full method achieves the highest VBench, Quality, and Semantic scores in Table 2. The SRDiffusion criterion scores 0.8288 VBench versus 0.8348 for EcoVideo++. Replacing all three decisions with fixed routing reduces VBench to 0.7972. Thus, the choice of signals matters even when the allocation budgets are held fixed.

Sensitivity to the switching threshold. The threshold τ determines when the clean-latent change r_i is small enough to trigger joint denoising. With all other settings fixed, we vary τ from 0.10 to 0.25. The highest VBench score in this sweep is 0.8348 at $\tau = 0.15$, with a 2.93× speedup (Table D1). Raising τ to 0.25 switches earlier and increases speedup to 3.26×, but lowers VBench to 0.8122. We therefore use $\tau = 0.15$ as the default. The complete sweep and sensitivity analyses for the temporal and spatial budgets appear in Appendix D.

6 Conclusion

Our analysis shows that large–small VDM capability gaps vary across Semantics Depth, Temporal Coverage, and Spatial Objects. Guided by this finding, EcoVideo++ adaptively transitions from full-video large-model denoising to joint denoising. During joint denoising, the small model predicts the full video, while large-model predictions for selected regions are computed with surrounding context and smoothly fused into it. Across Wan2.1, Wan2.2, and CogVideoX, EcoVideo++ achieves 2.29–2.93× end-to-end speedups over large-only inference, with VBench scores at most 0.0027 lower.

AI USE STATEMENT

We used ChatGPT to assist with translation, manuscript drafting and editing, literature search, clarification of methodological descriptions, feedback on ablation designs, and the interpretation and presentation of experimental results. The authors are responsible for verifying all AI-assisted material and take full responsibility for the final text, methods, experimental results, references, and conclusions.

REFERENCES

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

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 *CVPR*, 2023b.

Jiayu Chen, Hengyi Zhang, Maoliang Li, Minyu Li, Zihao Zheng, Xuanzhe Liu, Guojie Luo, and Xiang Chen. EcoVideo: Entropy-orchestrated video generation paradigm in cloud-edge dynamics. *arXiv preprint arXiv:2606.30557*, 2026. URL https://arxiv.org/abs/2606.30557.

Shenggan Cheng, Yuanxin Wei, Lansong Diao, Yong Liu, Bujiao Chen, Lianghua Huang, Yu Liu, Wenyuan Yu, Jiangsu Du, Wei Lin, and Yang You. SRDiffusion: Accelerate video diffusion inference via sketching-rendering cooperation, 2025. URL https://arxiv.org/abs/2505.19151.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, volume 35, 2022.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *NeurIPS*, volume 38, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Kumara Kahatapitiya, Haozhe Liu, Sen He, Ding Liu, Menglin Jia, Chenyang Zhang, Michael S. Ryoo, and Tian Xie. Adaptive caching for faster video generation with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15240–15252, 2025. URL https://arxiv.org/abs/2411.02397.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. doi: 10.48550/arXiv.2412.03603. URL https://arxiv.org/abs/2412.03603.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7353–7363, 2025.

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. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5775–5787, 2022. URL https://arxiv.org/abs/2206.00927.

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. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=W49UjcpGxx.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *CVPR*, pp. 15762–15772, 2024.

NVIDIA et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. URL https://arxiv.org/abs/2501.03575.

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. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2mqb8bPHeb.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, pp. 32211–32252, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

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

Wan Team. Wan2.2. Official model and code release, 2025. URL https://github.com/Wan-Video/Wan2.2.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse VideoGen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68208–68224, 2025. URL https://proceedings.mlr.press/v267/xi25c.html.

Jiajian Xie, Shengyu Zhang, Zhou Zhao, Fan Wu, and Fei Wu. EC-Diff: Fast and high-quality edge-cloud collaborative inference for diffusion models, 2025. URL https://arxiv.org/abs/2507.11980.

Chenqian Yan, Songwei Liu, Hongjian Liu, Xurui Peng, Xiaojian Wang, Fangmin Chen, Lean Fu, and Xing Mei. Hybrid SD: Edge-cloud collaborative inference for stable diffusion models, 2024. URL https://arxiv.org/abs/2408.06646.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. In *NeurIPS*, volume 38, 2025a.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025b. URL https://arxiv.org/abs/2408.06072.

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.

Yuanhao Zhai, Kevin Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, David Doermann, Junsong Yuan, and Lijuan Wang. Motion consistency model: Accelerating video diffusion with disentangled motion-appearance distillation. *arXiv preprint arXiv:2406.06890*, 2024.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *ICML*, pp. 74714–74731, 2025.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *ICLR*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *ICLR*, 2025.

A Related Work

Video Diffusion Models. Video diffusion extends image denoising architectures to jointly model appearance and temporal coherence (Ho et al., 2022). One line adapts pretrained image generators through temporal layers and staged video training, as in Video LDM and Stable Video Diffusion (Blattmann et al., 2023b;a). Latent diffusion reduces representation cost (Rombach et al., 2022), while Diffusion Transformers (DiTs) provide scalable denoisers (Peebles & Xie, 2023). Open-Sora, CogVideoX, Wan, and HunyuanVideo combine compressed spatiotemporal representations with transformer backbones and large-scale training (Zheng et al., 2024; Yang et al., 2025b; Wan et al., 2025; Kong et al., 2024). These developments improve video representations and generative capacity, but inference still repeatedly processes long token sequences. Pretrained variants with different parameter counts provide the basis for sharing this computation between large and small models.

Collaborative Video Diffusion. Collaborative diffusion distributes generation across models with different capacities. One line partitions the denoising trajectory into stages, as explored in image diffusion by T-Stitch and HybridSD (Pan et al., 2025; Yan et al., 2024). For video generation, SRDiffusion and EC-Diff use adaptive switching or handoff search to balance quality and computation through a large-to-small transition (Cheng et al., 2025; Xie et al., 2025). Another line partitions work across frames: EcoVideo assigns selected keyframes to a large model and reconstructs intermediate frames with a lightweight model through motion-aware interpolation (Chen et al., 2026). These approaches reserve expensive computation for selected stages or frames. Guided by tri-dimensional capability analysis, EcoVideo++ jointly selects the denoising stages, temporal segments, and spatial object regions that receive large-model computation. During joint large–small denoising, it smoothly fuses full-video small-model predictions with local large-model predictions that preserve spatiotemporal context.

Efficient Video Diffusion. Diffusion acceleration reduces the number or cost of denoising evaluations. Fast samplers shorten the inference trajectory (Song et al., 2021; Lu et al., 2022), while distillation and consistency learning produce generators that require fewer steps (Salimans & Ho, 2022; Song et al., 2023). Video-specific work addresses motion preservation through distillation (Zhai et al., 2024) and low-latency streaming through few-step autoregressive generation (Yin et al., 2025; Huang et al., 2025). A second line reuses intermediate computation across steps, from hierarchical features in image diffusion (Ma et al., 2024) to attention and feature outputs in video diffusion (Zhao et al., 2025; Lv et al., 2025). Refresh policies can adapt to predicted output changes, video content, or token sensitivity (Liu et al., 2025; Kahatapitiya et al., 2025; Zou et al., 2025). Sparse attention instead reduces interactions within each evaluation by exploiting spatiotemporal attention patterns and locality (Xi et al., 2025; Zhang et al., 2025), or by grouping semantically similar tokens (Yang et al., 2025a). These approaches improve sampling and computation within a generator; model collaboration allocates work across generators of different capacities. We examine their compatibility by combining EcoVideo++ with TeaCache in Appendix F.

B Implementation Details

Generation settings. We summarize the model pairs, resolutions, video lengths, and sampling steps in Table B1. All methods use the same generation settings within each model pair, following the evaluation protocol in Section 5.1.

Table B1: Model pairs and generation settings used in the main experiments.

Setting	Wan2.1 (2025)	Wan2.2 (2025)	CogVideoX (2025b)
Large model	Wan2.1-14B	Wan2.2-T2V-A14B	CogVideoX-5B
Small model	Wan2.1-1.3B	Wan2.1-1.3B	CogVideoX-2B
Resolution	1280×720	1280×720	720×480
Frames	81	81	49
FPS	16	16	8
Denoising steps	50	40	50

Algorithm 1 summarizes the inference procedure in Section 4. The large and small models operate in a shared latent space, and each denoising step produces one full-video prediction for the sampling scheduler. We use $i = 1, \ldots, N$ to index the execution order, so z_i is the latent before step i and z_{i+1} is its updated value. The algorithm uses the flow-matching parameterization of Wan (Wan et al., 2025), for which $\widehat{z}^L_{0,i} = z_i - \sigma_i \widetilde{u}^L_i$. Other prediction parameterizations use the corresponding scheduler conversion to obtain the clean-latent estimate.

Stage transition and prediction convention. During full-video large-model denoising, the large model processes the complete latent. We compute the relative change between consecutive clean-latent estimates using Eq. (4) and maintain a counter of consecutive values at or below τ. A value above τ resets the counter. Once the counter reaches P, the next denoising step enters the joint large–small denoising stage, which continues for the remaining steps. The first clean-latent estimate initializes the comparison and does not count as a stable step. All fused predictions use the same classifier-free guidance convention (Ho & Salimans, 2022), $\widetilde{u}^M_i = u^{M,\varnothing}_i + w(u^{M,c}_i - u^{M,\varnothing}_i)$, where $M \in \{L, S\}$ and w is the guidance scale. The spatial routing signal is computed from the conditional and unconditional predictions before this combination.

Initializing and updating the routing cache. Spatial selection uses the small-model CFG-gap map defined in Eq. (8). At the first joint denoising step $i = i_{\mathrm{sw}}$, we evaluate the small model on the current latent z_i and timestep t_i to initialize G_i, and use this map to select the local regions for the same step. Subsequent joint denoising steps construct their regions using the cached G_{i-1}. Each joint denoising step caches its current G_i for the next step.

Selecting temporal segments and spatial object regions. At each joint denoising step, we compute the latent-frame scores in Eq. (6), select the top ρ_f fraction of frames, and merge adjacent selected frames into temporal segments. The Wan implementation retains at most two temporal segments, ranked by their mean frame score. Within each temporal segment, the spatial router aggregates the CFG-gap map over frames and selects high-scoring spatial locations according to ρ_s. A 3×3 average filter smooths the score map, and a 3×3 neighborhood expands the selection mask before connected-component extraction. The bounding box of the component with the largest summed score defines the spatial core. Combining this box with its temporal segment gives one core region, with at most two cores per step. These operations convert scattered high-score locations into compact inputs for the large model. The selection fractions are applied before bounding-box construction and context expansion, so they do not directly specify the final number of tokens processed by the large model.

Context and position alignment. Each core region is expanded by one latent frame temporally and four latent positions along each spatial axis in the Wan implementation. Crop bounds are clipped to the full latent and aligned to the model's patch grid, with a minimum spatial crop size of 8×8 latent positions. We retain both the global crop origin and the core's coordinates within the crop.

Algorithm 1 EcoVideo++ inference

Require: Large model L, small model S, text condition c, guidance scale w, initial latent z_1, sampling schedule $\{(t_i, \sigma_i)\}_{i=1}^{N}$, threshold τ, patience P, budgets ρ_f, ρ_s
Ensure: Generated video x

stage $\leftarrow$ large, $q \leftarrow 0$, $\hat{z}_{\text{prev}} \leftarrow \varnothing$, $G_{\text{cache}} \leftarrow \varnothing$
for $i = 1, \ldots, N$ **do**
  if stage $=$ large **then**
    $(u_i^{L,c}, u_i^{L,\varnothing}) \leftarrow L(z_i, t_i, c),\ L(z_i, t_i, \varnothing)$
    $\widetilde{u}_i^{\text{out}} \leftarrow u_i^{L,\varnothing} + w(u_i^{L,c} - u_i^{L,\varnothing})$
    $\widehat{z}_{0,i}^{L} \leftarrow z_i - \sigma_i \widetilde{u}_i^{\text{out}}$
    if $\hat{z}_{\text{prev}} \neq \varnothing$ **then**
      $r_i \leftarrow \text{RMS}(\widehat{z}_{0,i}^{L} - \hat{z}_{\text{prev}})/(\text{RMS}(\hat{z}_{\text{prev}}) + \varepsilon)$
      Set $q \leftarrow q + 1$ if $r_i \leq \tau$, and $q \leftarrow 0$ otherwise
    end if
    $\hat{z}_{\text{prev}} \leftarrow \widehat{z}_{0,i}^{L}$
    if $q \geq P$ and $i < N$ **then**
      $i_{\text{sw}} \leftarrow i + 1$, stage $\leftarrow$ joint {effective from step $i + 1$}
    end if
  else
    if $G_{\text{cache}} = \varnothing$ **then**
      $(u_i^{S,c}, u_i^{S,\varnothing}) \leftarrow S(z_i, t_i, c),\ S(z_i, t_i, \varnothing)$
      $G_{\text{cache}} \leftarrow G_i$ computed using Eq. (8)
    end if
    Compute $s_{i,f}$ from z_i using Eq. (6)
    Select the top ρ_f fraction of frames and merge adjacent selections into temporal segments
    $\mathcal{I}_i \leftarrow$ retain up to two temporal segments ranked by their mean frame score
    $\mathcal{C}_i \leftarrow$ construct core regions from $\mathcal{I}_i$, G_{cache}, and ρ_s
    if $i > i_{\text{sw}}$ **then**
      $(u_i^{S,c}, u_i^{S,\varnothing}) \leftarrow S(z_i, t_i, c),\ S(z_i, t_i, \varnothing)$
    end if
    $\widetilde{u}_i^{S} \leftarrow u_i^{S,\varnothing} + w(u_i^{S,c} - u_i^{S,\varnothing})$; $\widetilde{u}_i^{\text{out}} \leftarrow \widetilde{u}_i^{S}$
    for each core region $C_{i,k} \in \mathcal{C}_i$ **do**
      Expand $C_{i,k}$ to a context crop $\widetilde{C}_{i,k}$ and obtain its global patch origin $o_{i,k}$
      Evaluate L on $z_i|_{\widetilde{C}_{i,k}}$ with CFG and RoPE coordinates $p = o_{i,k} + u$
      Extract the local large-model prediction for the core region $\widetilde{u}_{i,k}^{L,\text{local}}$
      $\widetilde{u}_i^{\text{out}}|_{C_{i,k}} \leftarrow \widetilde{u}_i^{S}|_{C_{i,k}} + \alpha_{i,k} \odot (\widetilde{u}_{i,k}^{L,\text{local}} - \widetilde{u}_i^{S}|_{C_{i,k}})$
    end for
    Compute G_i using Eq. (8) and cache $G_{\text{cache}} \leftarrow G_i$
  end if
  $z_{i+1} \leftarrow \text{SchedulerStep}(z_i, \widetilde{u}_i^{\text{out}}, t_i)$
end for
$x \leftarrow \text{VAEDecode}(z_{N+1})$
return x

The global origin is converted to patch coordinates and added to local token coordinates when constructing 3D RoPE (Su et al., 2021), retaining the original video's spatiotemporal coordinates as described in Section 4.4. After the large-model forward pass, we extract only the local large-model prediction for the core region. The spatiotemporal context contributes to that prediction but is not written back.

Smooth fusion and latent update. Starting from the full-video small-model prediction, we smoothly fuse each local large-model prediction into its corresponding core region according to Eq. (9). The Wan implementation constructs the fusion weight as a product of one-dimensional ramps along the temporal, height, and width axes. For an axis of length n and fusion width b, the ramp at index $j \in \{0, \ldots, n-1\}$ is $\min\{1, (j+1)/(b+1), (n-j)/(b+1)\}$. It is set to one when $b = 0$ or $n \leq 2b$. The respective fusion widths are 1, 2, and 2 latent positions. The resulting weight is largest in the interior and decreases toward the core boundary. Locations outside the core regions retain the small-model prediction. Finally, the scheduler advances the entire latent once using the fused prediction, and the VAE decodes the latent after the last denoising step.

B.1 Capability Analysis Protocol

Prompts and trajectories. The analyses in Figures 2 and 3 use 40 prompts, comprising 20 Simple and 20 Complex prompts. Simple prompts describe a single subject performing a single action, without explicit interactions between multiple subjects. Complex prompts involve interactions between multiple subjects, compound actions, or explicit spatial relations. We use three seeds per prompt, yielding 120 trajectories per model pair. Figure 2(a) presents the results for Wan2.1 and CogVideoX. For each model pair, we follow the large-only denoising trajectory and evaluate both models at the same latent state, timestep, text condition, and CFG settings.

Aggregation and uncertainty. For each prompt, we first average measurements over its three seeds. We then average across prompts, assigning equal weight to each prompt within the group being summarized. Confidence intervals are estimated by resampling prompts, keeping the seed-level measurements of each sampled prompt together.

C Full VBench Results

Tables C1–C3 report all 16 VBench dimensions (Huang et al., 2024), together with the Quality, Semantic, and overall VBench scores, for the three model pairs in Section 5.1. The model checkpoints, generation settings, and evaluation protocol follow the main experiments. Quality summarizes visual quality, while Semantic measures text–video alignment. The overall VBench score combines these aggregates with weights of 0.8 and 0.2, respectively. Bold values identify the best accelerated method in each column, with large-only inference shown as a reference.

EcoVideo++ achieves the highest overall VBench and Quality scores among the evaluated accelerated methods across all three model pairs, leading in 14 of 16 individual dimensions on each Wan backbone and 13 on CogVideoX; the remaining dimensions exhibit method-specific trade-offs.

Table C1: Full VBench results for Wan2.1-14B paired with Wan2.1-1.3B. Higher is better for all metrics. Bold indicates the best accelerated method in each column, excluding large-only inference.

Method	Subj. Cons.↑	Back. Cons.↑	Temporal Flicker.↑	Motion Smooth.↑	Dynamic Degree↑	Aesthetic Quality↑
Large-only	0.9627	0.9684	0.9908	0.9752	0.6813	0.6347
T-Stitch (Pan et al., 2025)	0.9102	0.9138	0.9408	0.9219	0.6213	0.5665
HybridSD (Yan et al., 2024)	0.7781	0.7862	0.8187	0.7826	0.4320	0.4768
SRDiffusion (Cheng et al., 2025)	0.9342	0.9416	0.9567	0.9491	0.6514	0.6038
EC-Diff (Xie et al., 2025)	0.7978	0.8114	0.8384	0.7967	0.4561	0.4920
EcoVideo++	**0.9589**	**0.9652**	**0.9857**	**0.9728**	**0.6985**	**0.6527**

Method	Imaging Quality↑	Object Class↑	Multiple Objects↑	Human Action↑	Color↑	Spatial Relation↑
Large-only	0.7216	0.9036	0.7128	0.9540	0.8859	0.6649
T-Stitch	0.6631	0.8478	0.6472	0.8773	0.8623	0.5984
HybridSD	0.5514	0.6905	0.4523	0.7000	0.6323	0.4211
SRDiffusion	0.6872	0.8734	0.6815	0.9461	**0.8896**	0.6327
EC-Diff	0.5679	0.7233	0.2331	0.6371	0.7795	0.3891
EcoVideo++	**0.7384**	**0.9142**	**0.7294**	**0.9483**	0.8816	**0.6798**

Method	Scene↑	Appearance Style↑	Temporal Style↑	Overall Cons.↑	Quality↑	Semantic↑	VBench↑
Large-only	0.7426	0.2264	0.2415	0.9218	0.8559	0.7611	0.8369
T-Stitch	0.6896	0.2073	0.2066	0.8690	0.8184	0.7195	0.7986
HybridSD	0.5299	0.2232	0.1817	0.7393	0.7239	0.4867	0.6765
SRDiffusion	0.7118	**0.2258**	0.2261	0.8875	0.8387	**0.7642**	0.8238
EC-Diff	0.5572	0.2253	0.1821	0.7566	0.7296	0.4976	0.6832
EcoVideo++	**0.7568**	0.2249	**0.2468**	**0.9326**	**0.8551**	0.7535	**0.8348**

Table C2: Full VBench results for Wan2.2-T2V-A14B paired with Wan2.1-1.3B. Higher is better for all metrics. Bold indicates the best accelerated method in each column, excluding large-only inference.

Method	Subj. Cons.↑	Back. Cons.↑	Temporal Flicker.↑	Motion Smooth.↑	Dynamic Degree↑	Aesthetic Quality↑
Large-only	0.9716	0.9741	0.9921	0.9813	0.7016	0.6578
T-Stitch	0.9167	0.9186	0.9473	0.9269	0.6263	0.5792
HybridSD	0.9003	0.8967	0.9309	0.9091	0.6085	0.5573
SRDiffusion	0.9436	0.9481	0.9638	0.9542	0.6628	0.6127
EC-Diff	0.9169	0.9188	0.9375	0.9221	0.6315	0.5794
EcoVideo++	**0.9689**	**0.9702**	**0.9886**	**0.9788**	**0.7196**	**0.6715**

Method	Imaging Quality↑	Object Class↑	Multiple Objects↑	Human Action↑	Color↑	Spatial Relation↑
Large-only	0.7348	0.9236	0.7614	0.9640	0.8743	0.7042
T-Stitch	0.6681	0.8622	0.6616	0.8823	0.8564	0.6128
HybridSD	0.6479	0.8427	0.6366	0.9000	0.8283	0.6033
SRDiffusion	0.7025	0.8845	0.6967	0.9517	**0.8832**	0.6468
EC-Diff	0.6733	0.8624	0.6718	0.9000	0.8583	0.6230
EcoVideo++	**0.7513**	**0.9317**	**0.7781**	**0.9572**	0.8684	**0.7184**

Method	Scene↑	Appearance Style↑	Temporal Style↑	Overall Cons.↑	Quality↑	Semantic↑	VBench↑
Large-only	0.7827	0.2039	0.2187	0.9318	0.8542	0.7950	0.8423
T-Stitch	0.6961	0.2094	0.2210	0.8834	0.8276	0.7314	0.8084
HybridSD	0.6821	**0.2304**	0.2139	0.8615	0.8001	0.7183	0.7837
SRDiffusion	0.7314	0.2051	0.2241	0.9018	0.8418	0.7665	0.8267
EC-Diff	0.7063	0.2239	0.2212	0.8736	0.8042	0.7431	0.7920
EcoVideo++	**0.8017**	0.2026	**0.2364**	**0.9384**	**0.8538**	**0.7828**	**0.8396**

Table C3: Full VBench results for CogVideoX-5B paired with CogVideoX-2B. Higher is better for all metrics. Bold indicates the best accelerated method in each column, excluding large-only inference.

Method	Subj. Cons.↑	Back. Cons.↑	Temporal Flicker.↑	Motion Smooth.↑	Dynamic Degree↑	Aesthetic Quality↑
Large-only	0.9254	0.9281	0.9617	0.9462	0.6384	0.5937
T-Stitch	0.8631	0.8571	0.9025	0.8828	0.5734	0.5165
HybridSD	0.8329	0.8386	0.8723	0.8420	0.4926	0.4980
SRDiffusion	0.8792	0.8894	0.9168	0.9021	0.5897	0.5486
EC-Diff	0.9072	0.9080	0.9521	**0.9542**	**0.8333**	0.4855
EcoVideo++	**0.9218**	**0.9246**	**0.9584**	0.9417	0.6518	**0.6124**

Method	Imaging Quality↑	Object Class↑	Multiple Objects↑	Human Action↑	Color↑	Spatial Relation↑
Large-only	0.6815	0.8518	0.6572	0.8600	0.9203	0.6134
T-Stitch	0.6116	0.7876	0.5937	0.7784	0.8307	0.5370
HybridSD	0.5908	0.7405	0.4987	0.6400	0.7957	0.4699
SRDiffusion	0.6328	0.8126	0.6127	0.8538	0.9016	0.5683
EC-Diff	0.5655	0.6094	0.2942	0.7000	0.8112	0.4232
EcoVideo++	**0.6973**	**0.8674**	**0.6729**	**0.8567**	**0.9148**	**0.6278**

Method	Scene↑	Appearance Style↑	Temporal Style↑	Overall Cons.↑	Quality↑	Semantic↑	VBench↑
Large-only	0.6918	0.2280	0.2316	0.8837	0.7964	0.7064	0.7784
T-Stitch	0.6313	0.2222	0.2064	0.8119	0.7677	0.6686	0.7479
HybridSD	0.5711	0.2183	0.1893	0.7917	0.7772	0.5178	0.7253
SRDiffusion	0.6517	**0.2319**	0.2164	0.8346	0.7895	**0.6960**	0.7708
EC-Diff	0.1453	0.2251	0.2012	0.2281	0.7626	0.5535	0.7208
EcoVideo++	**0.7086**	0.2308	**0.2382**	**0.8972**	**0.8012**	0.6797	**0.7769**

D SENSITIVITY TO SWITCHING AND SPATIOTEMPORAL BUDGETS

Switching threshold. We vary τ from 0.10 to 0.25 while keeping the remaining configuration fixed to study the transition from full-video large-model denoising to joint large–small denoising. A larger threshold relaxes the clean-latent stability criterion and permits an earlier transition. Increasing τ from 0.10 to 0.15 reduces latency from 886.75 to 674.33 seconds while slightly improving VBench from 0.8336 to 0.8348 (Table D1). Further increases provide additional acceleration but reduce quality: $\tau = 0.20$ and 0.25 achieve $3.01\times$ and $3.26\times$ speedups, with VBench scores of 0.8232 and 0.8122. We therefore use $\tau = 0.15$ for the main comparison, as it gives the highest VBench score in this sweep together with a $2.93\times$ speedup.

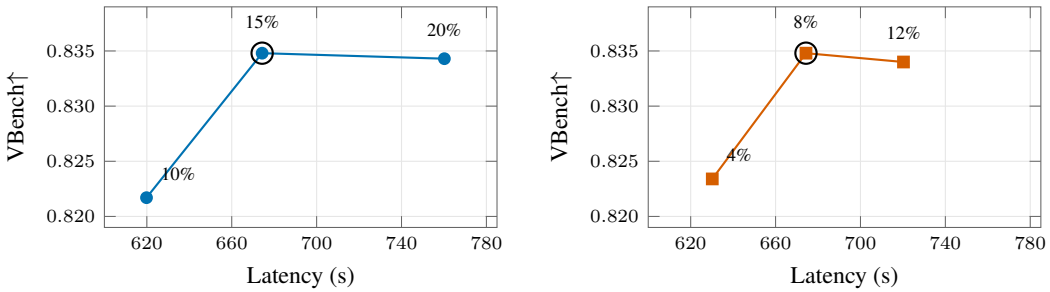
Table D1: Sensitivity to the dynamic switching threshold τ.

τ	VBench ↑	Quality ↑	Semantic ↑	Latency (s) ↓	Speedup ↑
Large-only	0.8369	0.8559	0.7611	1977.86	1.00×
0.10	0.8336	0.8543	0.7508	886.75	2.23×
0.15	**0.8348**	**0.8551**	**0.7535**	674.33	2.93×
0.20	0.8232	0.8419	0.7486	656.58	3.01×
0.25	0.8122	0.8328	0.7297	**607.15**	**3.26×**

We vary the frame budget ρ_f and spatial budget ρ_s on Wan2.1-14B paired with Wan2.1-1.3B to examine how the extent of local large-model computation affects generation quality and latency. The frame budget specifies the fraction of latent frames selected at each joint denoising step. The spatial budget specifies the fraction of high-scoring spatial locations selected within each temporal segment, before region construction and context expansion. Figure D1 presents two sweeps, each varying one budget while holding the other budget and all remaining settings fixed.

Frame budget. With $\rho_s = 8\%$, increasing ρ_f from 10% to 15% raises VBench from 0.8217 to 0.8348, while latency increases from 619.84 to 674.33 seconds. Increasing ρ_f further to 20% requires 760.19 seconds and yields a VBench score of 0.8343. Thus, expanding temporal coverage improves quality from the lowest tested budget, but the additional computation beyond 15% does not produce a higher VBench score in this sweep.

Spatial budget. At $\rho_f = 15\%$, increasing ρ_s from 4% to 8% improves VBench from 0.8234 to 0.8348 and increases latency from 630.17 to 674.33 seconds. A spatial budget of 12% takes 720.27 seconds and obtains a score of 0.8340. Both sweeps therefore favor the default combination of $\rho_f = 15\%$ and $\rho_s = 8\%$: it achieves the highest VBench score among the tested configurations, while either larger budget adds latency without improving that score. The selected fractions govern routing, whereas the actual large-model input also includes the surrounding context described in Appendix B.



(a) Frame budget (b) Spatial budget

Figure D1: Spatiotemporal budget sensitivity on Wan2.1-14B paired with Wan2.1-1.3B. (a) Frame-budget sweep with $\rho_s = 8\%$. (b) Spatial-budget sweep with $\rho_f = 15\%$. Labels show the varied budget, and black circles mark the default configuration. Each point is an evaluated configuration, with latency measured in seconds.

E MORE QUALITATIVE RESULTS

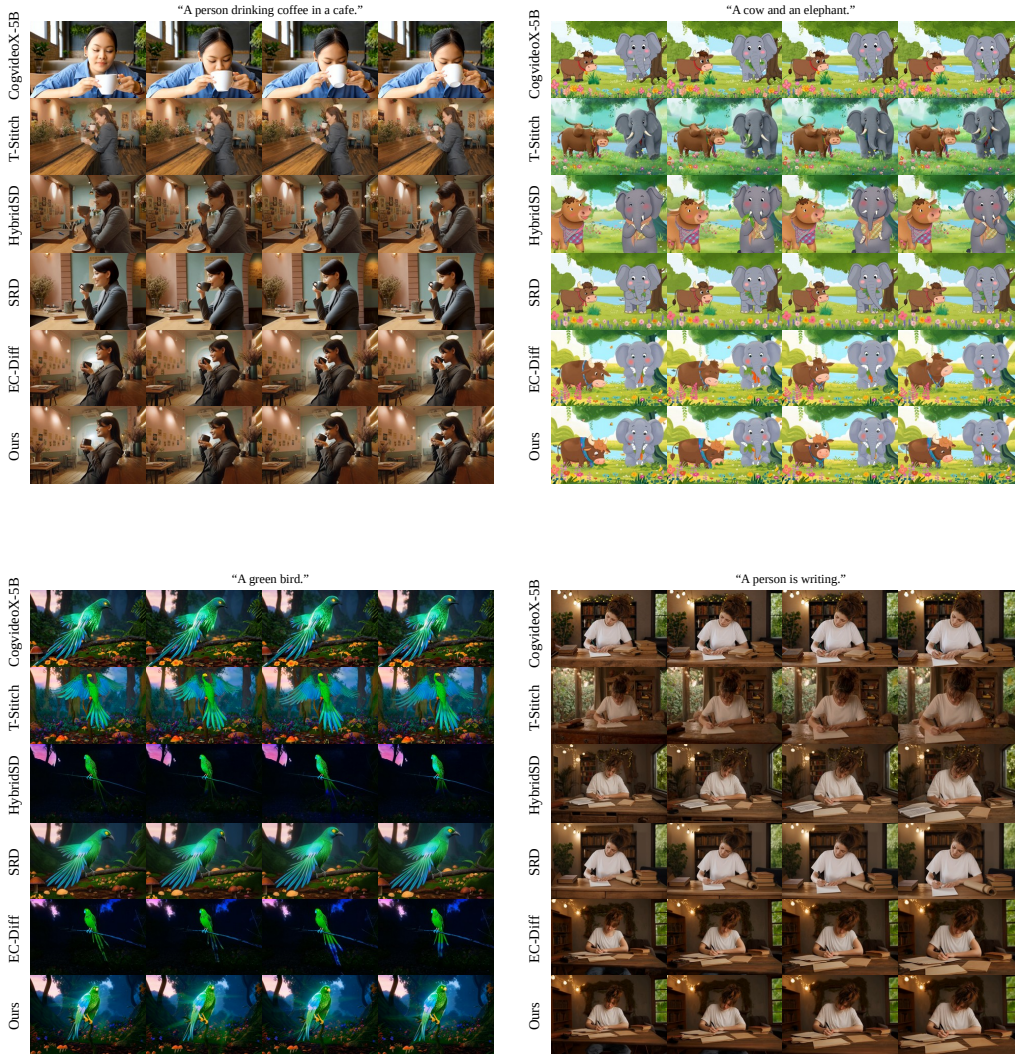


Figure E2: Qualitative results of CogVideoX.

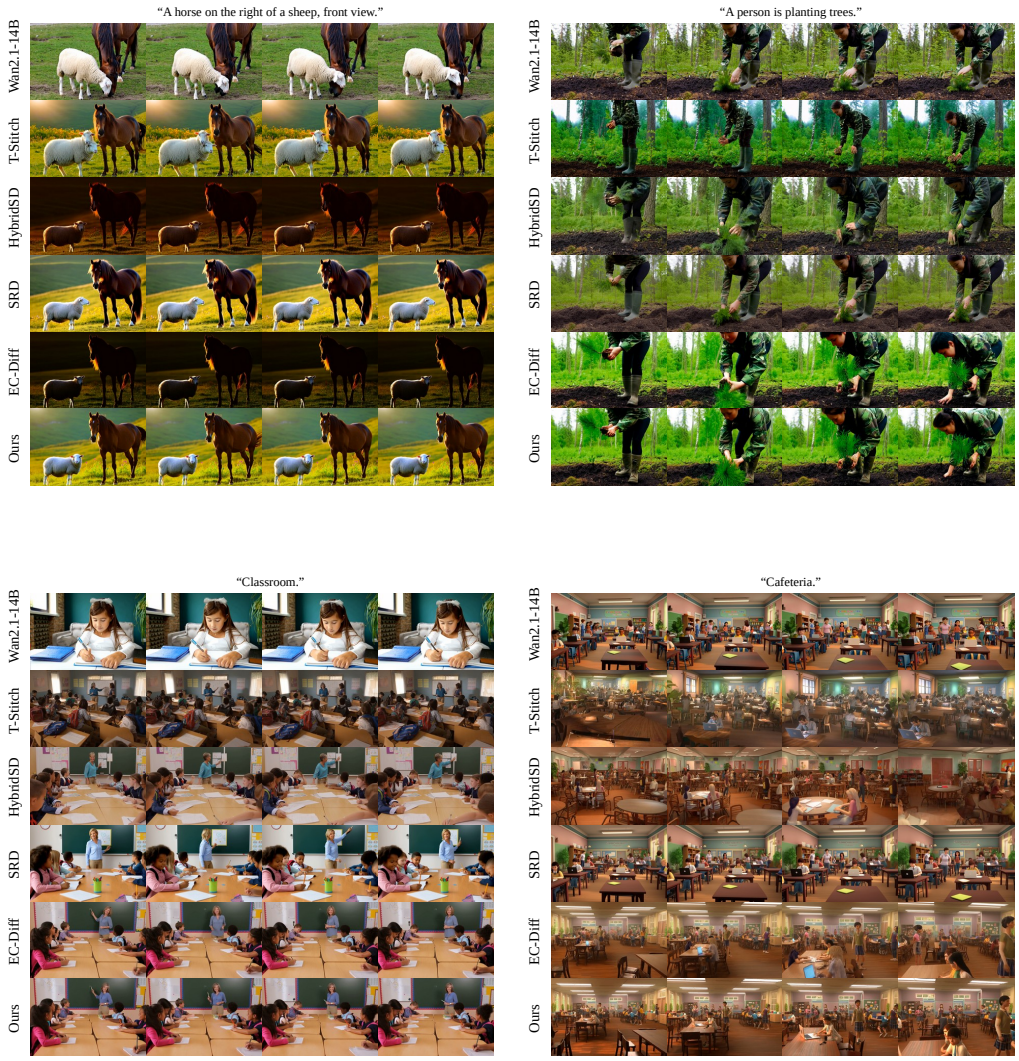


Figure E3: Qualitative results of Wan2.1.

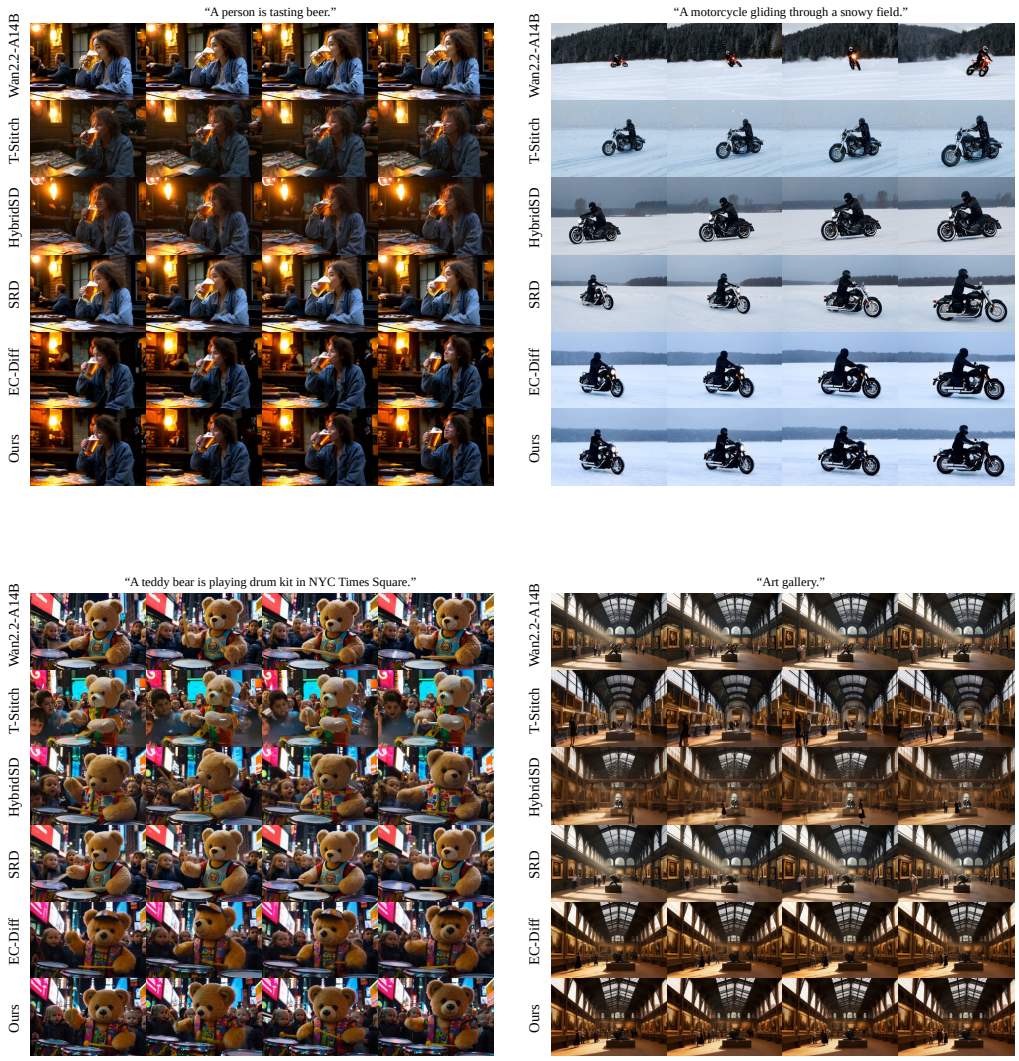


Figure E4: Qualitative results of Wan2.2.

F COMPATIBILITY WITH TEACACHE

EcoVideo++ allocates computation between large and small models, while TeaCache (Liu et al., 2025) reuses intermediate model outputs across denoising steps. We examine their combination on Wan2.1-14B paired with Wan2.1-1.3B. Table F1 compares large-only inference and EcoVideo++, each with and without TeaCache, to show how caching changes the quality–latency trade-off of both inference strategies. The speedups reported in Table F1 use large-only inference without caching as the reference.

Evaluation protocol. All four configurations follow the full evaluation protocol in Section 5.1. We use the complete VBench prompt set with five seeds per prompt, shared across methods, and report mean scores. The overall VBench score incorporates all 16 dimensions through the Quality and Semantic aggregates, weighted by 0.8 and 0.2, respectively. Each video contains 81 frames at 1280×720 resolution and 16 FPS, generated with 50 denoising steps. EcoVideo++ uses the default switching threshold $\tau = 0.15$. Experiments run on a single NVIDIA H200 GPU with batch size 1. Reported latency is measured end to end, including prompt encoding, the complete denoising process, and VAE decoding, while excluding model loading and disk I/O.

Table F1: Combining EcoVideo++ with TeaCache (Liu et al., 2025) on Wan2.1-14B/1.3B. Speedup is relative to large-only inference. Bold marks the highest VBench score and the lowest latency among accelerated configurations.

Method	VBench↑	Latency (s)↓	Speedup↑
Large-only	0.8369	1977.86	1.00×
Large-only + TeaCache	0.8207	1222.22	1.62×
EcoVideo++	**0.8348**	674.33	2.93×
EcoVideo++ + TeaCache	0.8189	**504.79**	**3.92**×

With TeaCache enabled for both methods, EcoVideo++ achieves a $2.42\times$ speedup over the cached large-only baseline, with a VBench score 0.0018 lower. Relative to EcoVideo++ without caching, TeaCache reduces latency by 25.14%, changing VBench from 0.8348 to 0.8189.

G LIMITATIONS

EcoVideo++ currently assumes that the large and small models operate in a shared latent space; collaboration across different latent representations remains unexplored. Temporal segments and spatial object regions are selected adaptively, while the frame and spatial budgets are predefined and control the quality–efficiency trade-off. Future work could extend collaboration across latent spaces and adjust these budgets automatically according to video content and available computation.