REPLANFLOW: ADAPTIVE FLOW-MAP SCHEDULING FOR FEW-STEP VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step video generation depends on both the accuracy of learned transitions and their composition under a limited inference budget. Finite-time flow maps allow flexible sampling endpoints, but preset endpoints during on-policy distillation and inference limit video quality when only a few sampling steps are available. We present ReplanFlow, which combines broader on-policy transition coverage with adaptive flow-map scheduling. During distillation, we sample nonuniform internal endpoints to prepare the generator for more varied transition compositions. Meanwhile, we train a lightweight endpoint predictor offline using quality scores of fully generated videos. At inference, it selects the next sampling endpoint based on the current video latent, text condition, and remaining step budget, using exactly the specified number of sampling steps. This requires neither candidate-video search nor online quality evaluation. On VBench, ReplanFlow outperforms the AnyFlow baseline at the same sampling budget under matched generator-training data. With only three sampling steps, it also surpasses the four-step baseline while achieving a measured $1.21\times$ end-to-end speedup. Component ablations further show that broader training coverage increases the gains from adaptive scheduling.

1 INTRODUCTION

Video diffusion models require repeated forward passes through large neural networks, making generation computationally expensive. Distillation reduces this cost by enabling generation in fewer sampling steps (Yin et al., 2024a; Huang et al., 2025). Flow maps provide a flexible foundation for few-step generation by learning transitions between arbitrary time points along a probability-flow ODE trajectory (Kim et al., 2024). Each model call can advance to a chosen target time, allowing transitions of different lengths to be composed within a given sampling budget. AnyFlow demonstrates the effectiveness of this formulation for video generation across different sampling budgets (Gu et al., 2026). However, its default sampling procedure follows a predefined time schedule. This leaves room to better exploit the flexibility of flow maps by choosing transitions according to the generation state and available budget.

This flexibility raises a further question: *how should a limited number of sampling steps be allocated along the trajectory?* A learned flow map only approximates the underlying transitions, so different time schedules can produce different results at the same sampling budget. Prior work on diffusion sampling has shown that schedule choice can substantially affect generation quality (Sabour et al., 2024). For video flow maps, this motivates studying both the transitions used during training and their selection during inference.

AnyFlow already trains over broad time pairs in its first stage. However, its second-stage on-policy distillation samples internal intervals from a predefined grid for each sampled step count (Gu et al., 2026). Its default inference procedure also follows a predefined schedule, regardless of the evolving video latent. The model therefore supports endpoint choices beyond those selected by its default sampling procedure. We investigate whether broader on-policy training coverage, together with state-dependent endpoint selection, can make better use of this flexibility under a small sampling budget.

We introduce ReplanFlow, guided by the principle that training coverage and inference scheduling should support each other: adaptive scheduling requires a generator trained on varied transition

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.

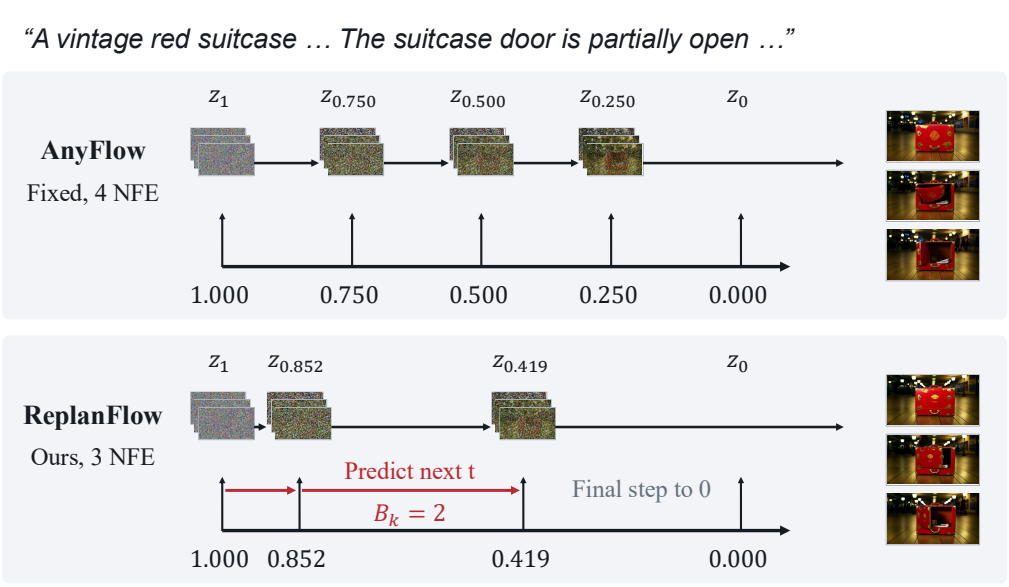

Figure 1: **Adaptive flow-map scheduling with fewer generator calls.** ReplanFlow selects endpoints from the evolving latent and remaining budget, using three calls instead of the AnyFlow baseline's four fixed calls. The final call reaches the clean endpoint. Times are shown in normalized pre-shift coordinates.

intervals, while broader coverage provides more endpoint choices for the scheduler to exploit. Endpoint selection should account for both the evolving generation state and its effect on final video quality. We therefore pair broader on-policy transition coverage with a lightweight predictor trained offline from final video quality to select endpoints as generation proceeds. On VBench (Huang et al., 2024), ReplanFlow improves quality at the same sampling budget and surpasses the four-step AnyFlow baseline with only three steps.

Our contributions are threefold:

- We broaden on-policy transition coverage through nonuniform internal endpoint sampling to support adaptive flow-map composition. Component ablations show larger scheduling gains with the resulting generator.
- We develop a state- and budget-conditioned predictor that selects flow-map endpoints sequentially under an exact generator-call budget. It learns offline from the terminal quality of candidate continuations, requiring neither candidate-video search nor quality evaluation at inference.
- We demonstrate improved video quality across sampling budgets under matched generator-training settings. Our three-step configuration surpasses the reproduced four-step AnyFlow baseline while reducing end-to-end generation time.

2 Related Work

Few-step video generation. Recent video diffusion models have substantially improved visual quality and temporal coherence (Yang et al., 2025; Team Wan et al., 2025), but their iterative sampling remains computationally expensive. Few-step generators use consistency training (Song et al., 2023; Lu & Song, 2025; Zheng et al., 2026), distribution matching (Yin et al., 2024b;a), or adversarial post-training (Lin et al., 2025). Salt improves low-NFE composition and cache robustness (Ge et al., 2026), while Parallel Decoding Distillation predicts multiple updates per network evaluation (Shaul et al., 2026). For causal video diffusion, CausVid distills a bidirectional teacher into a few-step causal student (Yin et al., 2025), while Self Forcing addresses exposure bias by training on self-generated contexts (Huang et al., 2025). DSA instead allocates a variable number of de-

noising steps across autoregressive frames using a jointly trained confidence head (Le et al., 2026). These methods primarily reduce or adapt denoising computation; we study how learned finite-time transitions can be composed under an exact generator-call budget.

Flow-map modeling and distillation. Flow maps model finite-time transitions, extending the instantaneous velocity formulation used in flow matching and rectified flow (Lipman et al., 2023; Liu et al., 2023). Consistency Trajectory Models learn transitions between arbitrary time pairs (Kim et al., 2024), and Flow Map Matching provides a mathematical framework for learning such two-time mappings (Boffi et al., 2025b). Flow-map self-distillation further systematizes Eulerian, Lagrangian, and progressive formulations (Boffi et al., 2025a), while related approaches learn transitions through step-size conditioning (Frans et al., 2025), interval-average velocities (Geng et al., 2025; 2026), or continuous-time distillation objectives (Sabour et al., 2025). In video generation, Transition Matching Distillation combines a shared backbone with a lightweight flow head for inner updates (Nie et al., 2026), while AnyFlow uses shortcut flow-map transitions for efficient on-policy distillation across sampling budgets (Gu et al., 2026). Our work builds on AnyFlow, whose first stage covers broad time pairs but whose second stage samples internal intervals from preset schedules. Continuous-Time Distribution Matching also explores randomized training schedules, together with an additional off-trajectory alignment objective for image distillation (Liu et al., 2026). We broaden the Stage-2 on-policy transition coverage of video flow maps and couple this training change with sequential scheduling under a fixed generator-call budget.

Sampling schedule optimization. The placement of sampling timesteps affects the quality attainable with a limited number of model evaluations. Prior work optimizes schedules through numerical error minimization (Xue et al., 2024), trajectory geometry (Chen et al., 2024), stochastic-calculus objectives (Sabour et al., 2024), and teacher-solver supervision (Tong et al., 2025). For flow-matching models, BOSS couples globally optimized Bellman stepsizes with velocity-field straightening (Nguyen et al., 2024). S4S learns solver coefficients and optionally a discretization grid (Frankel et al., 2025), while ART and HSO optimize model-level schedules (Huang et al., 2026; Zhu et al., 2026). Instance-dependent methods differ in when they select timesteps and what they control. INDIS predicts a complete fixed-budget schedule from the initial noise and condition by matching the output of a high-NFE teacher solver (Yuan et al., 2026); Yu et al. (2026) learn such initial plans from terminal rewards using REINFORCE. AMED uses current U-Net features to predict internal solver evaluation times and direction scaling (Zhou et al., 2024). Schedule On the Fly and RaSS also read intermediate states, while balancing generation quality against sampling cost (Ye et al., 2025; Ding et al., 2025). BFMT adapts flow-map transition times through online reward-guided tree search, incurring candidate-generation and scorer costs at inference (Ji et al., 2026). We instead select the next finite-time video flow-map endpoint under an exact generator-call budget and broaden on-policy training coverage to support these transitions. Our policy learns from offline terminal-quality evaluations and updates its decisions during generation without online candidate search or reward evaluation.

3 METHOD

As shown in Fig. 2, ReplanFlow combines broader on-policy transition coverage with budget-constrained endpoint selection. We first define scheduling under a fixed sampling budget, then describe the change to flow-map distillation. We next explain how terminal video quality supervises the endpoint predictor and how the predictor guides generation.

3.1 FLOW MAPS AND THE SCHEDULING PROBLEM

Finite-time transitions. Let z_t be a video latent at normalized time t, with noise at $t = 1$ and the clean endpoint at $t = 0$. Given a text condition c, a learned flow map performs a finite-time transition (Kim et al., 2024; Gu et al., 2026):

$$z_r = F_\theta(z_t, t, r, c), \qquad 0 \le r < t \le 1. \tag{1}$$

We use normalized pre-shift coordinates throughout; F_θ includes the fixed scheduler transformation to model times. Each transition requires one generator call. The learned map approximates the underlying flow, so composing different intermediate endpoints can change the final video.

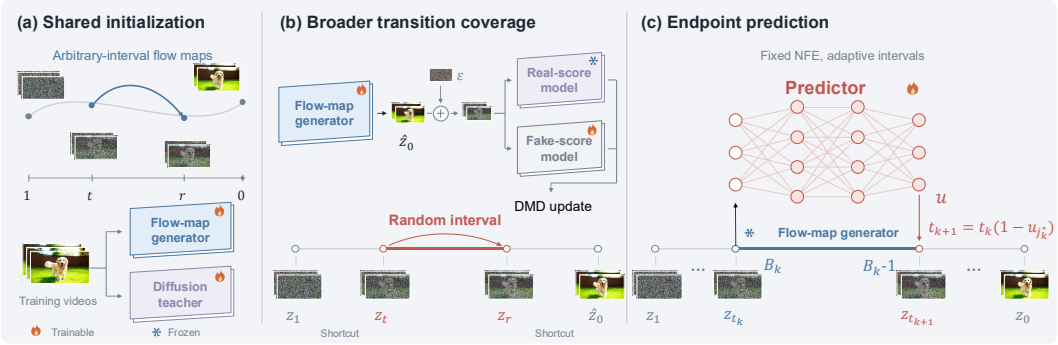

Figure 2: **Overview of ReplanFlow.** (a) Flow-map initialization and separately trained diffusion teacher. (b) Broader on-policy transition coverage through randomized internal endpoints and shortcut simulation. (c) Budget-constrained endpoint selection from the evolving generation state. Fire and snowflake symbols indicate trainable and frozen modules during their respective training stages. In (c), the predictor is trained offline using terminal-quality scores from same-prefix candidate continuations, with the generator frozen. The depicted endpoint execution is used at inference, when both the predictor and generator are frozen and no terminal-quality evaluator is required.

The scheduling objective. A budget of B sampling steps requires a decreasing sequence $1 = t_0 > t_1 > \cdots > t_B = 0$. At decision k, we represent the scheduling state as

$$s_k = (z_{t_k}, t_k, c, B_k, h_k), \qquad B_k = B - k, \qquad h_k = (t_0, \ldots, t_k). \tag{2}$$

Here, B_k counts remaining generator calls, including the next call, and h_k records the executed endpoints. When $B_k > 1$, a legal next endpoint satisfies $0 < r < t_k$; when $B_k = 1$, it must be $r = 0$. Our objective is to choose endpoints that improve the quality of the completed video while using exactly B generator calls, or NFE. A fixed schedule specifies the endpoints independently of the sample. An initial-only planner conditions its schedule on the initial noise and text. Our sequential predictor also observes the latent produced by each executed transition. Predictor computation is accounted for separately from generator NFE.

3.2 Broadening On-Policy Transition Coverage

Sampling internal endpoints. AnyFlow's Stage 1 learns transitions over broad time pairs, but its Stage 2 on-policy distillation selects internal intervals from a preset schedule for each sampled step count (Gu et al., 2026). In pre-shift coordinates, these endpoints are $a_i = 1 - i/N$ for a sampled count N. We broaden this training distribution by replacing the internal endpoints with sorted random samples (Fig. 2(b)):

$$\begin{aligned} a_0 = 1 > a_1 > \cdots > a_{N-1} > a_N = 0, \\ (a_1, \ldots, a_{N-1}) = \mathrm{sort}_{\downarrow}(U_1, \ldots, U_{N-1}). \end{aligned} \tag{3}$$

Here, $U_i \overset{\text{iid}}{\sim} \mathcal{U}(0,1)$. Both training branches use the same step-count distribution and scheduler shift. They also share the teacher and Stage 1 initialization. The change therefore concerns the intervals presented to on-policy distillation, rather than the flow-map parameterization.

Distilling the sampled transition. We sample an interval index i uniformly from $\{0, \ldots, N-1\}$ and set $t = a_i$, $r = a_{i+1}$. The student uses the original shortcut simulation to construct a terminal sample:

$$z_1 \xrightarrow{F_\theta} z_t \xrightarrow{F_\theta} z_r \xrightarrow{F_\theta} \widehat{z}_0. \tag{4}$$

Thus, the selected interval is executed on a state produced by the student itself. The terminal sample is re-noised at a separately sampled noise level. A frozen real-score model and a trainable fake-score model supply the distribution-matching update, following AnyFlow's DMD procedure (Yin et al., 2024b;a; Gu et al., 2026). The fake-score model is trained on generated samples. During the generator update, gradients propagate through all executed flow-map transitions in the shortcut

simulation, while the score-model predictions are treated as stop-gradient targets. The simulation uses at most three generator calls, skipping identity transitions at the boundaries; it does not unroll all N intervals.

Connection to inference. Nonuniform sampling exposes distillation to a wider range of transition intervals that an adaptive scheduler may compose. The trained generator can still use either fixed or adaptive inference. This separation allows the training change and endpoint selection to be evaluated independently in the component comparison. Appendix A visualizes training coverage and inference endpoint distributions.

3.3 Learning from Terminal Video Quality

Comparing candidate continuations. The value of an endpoint depends on the transitions that follow it. We train the endpoint predictor offline using terminal-quality scores of candidate videos produced by a frozen generator. For each prompt and noise seed, we enumerate a finite set of schedules on a discrete endpoint grid and reuse shared prefixes to collect states. For a state s with remaining budget b, candidate continuations share the same executed prefix and use exactly b further generator calls. Only endpoints that leave enough grid points for the remaining calls are included. The target for a candidate endpoint r is

$$\widehat{Q}_b(s, r) = \max_{\sigma \in \mathcal{C}_b(s,r)} R(y_\sigma, c), \tag{5}$$

where $\mathcal{C}_b(s, r)$ contains the enumerated continuations beginning with r, and R is a VBench-derived training quality score for the completed video y_σ (Huang et al., 2024). This target measures the best evaluated continuation within the candidate set, not a continuous-time optimum or the value of the deployed policy. States and labels are generated separately for each generator checkpoint.

Fitting an endpoint distribution. Several endpoints may have similar continuation scores. We use a categorical predictor to retain these alternatives during training. The predictor combines encoded latent, text, endpoint-history, and time/budget features with a compact attention module. It outputs $p_\phi(j \mid s)$ over $M = 64$ relative-progress bins, centered at $u_j = (j + 1/2)/M$. For candidate r_ℓ at current time t, let $u_\ell = 1 - r_\ell/t$ and $q_\ell = \widehat{Q}_b(s, r_\ell)$. We form a soft target and minimize cross-entropy:

$$\widetilde{p}_j(s) \propto \sum_\ell \exp\left(\frac{q_\ell - \max_m q_m}{\tau} - \frac{(u_j - u_\ell)^2}{2\sigma^2} \right), \tag{6}$$

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_s \sum_{j=0}^{M-1} \widetilde{p}_j(s) \log p_\phi(j \mid s). \tag{7}$$

The target is normalized over bins; τ controls the preference for higher scores and σ controls smoothing across nearby endpoints. We estimate the expectation over states using a depth-based curriculum. Training starts with initial states, then includes states after one transition, and finally covers all available decision depths. Minibatches are balanced across the active depths. Only predictor parameters are optimized in this stage. The reported configurations train separate predictors for each total sampling budget, while conditioning decisions on the remaining budget within a rollout.

3.4 Budget-Constrained Sequential Scheduling

Executing the learned schedule. Generation starts from $s_0 = (z_1, 1, c, B, (1))$. When $B_k > 1$, the predictor selects the most probable progress bin and the generator executes its endpoint (Fig. 2(c)):

$$\begin{gathered} j_k^* = \arg\max_j p_\phi(j \mid s_k), \qquad t_{k+1} = t_k(1 - u_{j_k^*}), \\ z_{t_{k+1}} = F_\theta(z_{t_k}, t_k, t_{k+1}, c). \end{gathered} \tag{8}$$

We append t_{k+1} to the history, decrement B_k, and feed the updated latent into the next decision. When one call remains, we set $t_{k+1} = 0$ without querying the predictor. All internal bin centers lie strictly between zero and one, so the schedule decreases and reaches zero in exactly B generator calls. The relative parameterization allows inference endpoints outside the grid used to construct supervision.

Online computation and feedback. Both the generator and predictor are frozen at inference. The procedure uses $B - 1$ predictor calls and requires no candidate-video search or terminal-quality evaluation online. Sequential scheduling can respond to the realized latent after each transition; an initial-only planner commits to its continuation without these observations. We distinguish the quality gain of the complete method from the contribution of this feedback. The latter is examined with a shared-first-transition control, which holds the first endpoint and resulting latent fixed before comparing planned and sequential continuations.

4 Experiments

4.1 Implementation Details

Models and training. We build ReplanFlow on Wan2.1-T2V-1.3B (Team Wan et al., 2025). With AnyFlow's original training data unavailable, we use approximately 4.3K public Wan-Syn pairs[1] instead of its reported 256K (Gu et al., 2026) for teacher adaptation and both training stages. Both AnyFlow and ReplanFlow use this same corpus and share the teacher and Stage 1 initialization; Stage 2 uses preset and sampled nonuniform endpoints, respectively. Both branches train with rank-256 LoRA for 1,200 updates at global batch size 16. We evaluate their EMA checkpoints.

Predictor and inference. Main predictors use 64 categorical bins and hidden width 128, trained separately per budget with five-seed supervision from each frozen generator. The two-, three-, and four-step predictors use 28.3K, 89.7K, and 136.9K state–action labels, respectively. We use AdamW for 5,000 updates, batch size 48, and initial learning rate 5×10^{-5}. Videos have 77 frames at 448×832 and 16 fps, with guidance scale 1 and scheduler shift 5. Fixed inference follows AnyFlow's default schedule; sequential inference uses the predictor. Each step uses one generator call; predictor overhead is measured separately. For literature entries, 50×2 denotes 50 sampling steps with two calls per step for classifier-free guidance.

Evaluation. For text-to-video evaluation, we adopt VBench (Huang et al., 2024), which measures generation quality across 16 dimensions grouped into Quality and Semantic scores. We report both scores and their aggregate Total score on a 0–100 scale. All score differences are computed before rounding. Main results and component ablations use VBench full with five generation seeds. The feedback analysis uses VBench medium, a 192-prompt subset evaluated with generation seed 0. Literature results are reported separately from our controlled comparisons. For Wan2.1 and rCM, we use the scores re-evaluated and reported by the AnyFlow authors (Gu et al., 2026).

4.2 Main Results

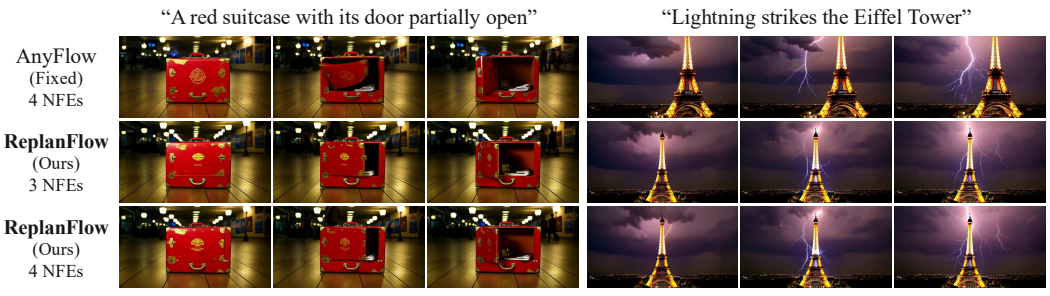

Figure 3: **Qualitative comparison with fixed-schedule AnyFlow.** Each example shows three video frames. Rows compare four-step AnyFlow with three- and four-step ReplanFlow, respectively.

Higher quality with fewer sampling steps. Table 1 compares ReplanFlow with AnyFlow under matched training settings. Fig. 5(a) shows the corresponding quality–budget tradeoff on VBench full. With three sampling steps, ReplanFlow achieves a VBench Total score of 83.34, exceeding

[1] https://huggingface.co/datasets/johnnyeric/Wan-Syn_77x448x832_600k

Table 1: **Video generation quality across sampling budgets.** Our controlled comparisons use VBench full.

Model	#Params	Resolution	NFEs ↓	Evaluation Scores ↑		
				Quality	Semantic	Total
Bidirectional — Literature results						
Wan2.1-T2V-1.3B	1.3B	480×832	50×2	84.99	76.23	83.24
rCM-Wan2.1-T2V-1.3B	1.3B	480×832	4	84.71	73.74	82.51
Bidirectional — Matched training settings						
AnyFlow (Fixed)	1.3B	448×832	2	84.08	74.01	82.07
AnyFlow (Fixed)	1.3B	448×832	3	84.82	75.20	82.89
AnyFlow (Fixed)	1.3B	448×832	4	85.03	75.42	83.11
ReplanFlow (Ours)	1.3B	448×832	2	84.24	75.88	82.57
ReplanFlow (Ours)	1.3B	448×832	3	85.14	76.14	83.34
ReplanFlow (Ours)	1.3B	448×832	4	**85.28**	**76.33**	**83.49**

the four-step AnyFlow baseline at 83.11. It improves both Quality and Semantic scores while using 25% fewer generator calls. At the same three-step budget, ReplanFlow improves the Total score by 0.45 points over AnyFlow. These results show that the flexibility of video flow maps can translate into better generation quality under a smaller sampling budget.

Improvement at a matched budget. ReplanFlow also improves quality at two and four sampling steps. At two steps, it achieves a Total score of 82.57, surpassing AnyFlow by 0.50 points. At four steps, it reaches 83.49, a gain of 0.39 points, with Quality and Semantic scores increasing by 0.25 and 0.91 points, respectively. The gains across two, three, and four sampling steps support the joint use of broader on-policy transition coverage and adaptive endpoint selection. Fig. 3 complements these scores with qualitative examples of object structure and prompt-specified actions, showing a suitcase opening and lightning striking the Eiffel Tower. Appendices B and E provide per-dimension scores and more examples with full prompts.

4.3 Ablation Studies

Training coverage and adaptive scheduling. Table 2 examines how broader on-policy transition coverage supports adaptive scheduling. At four sampling steps, sequential scheduling improves the Total score by 0.03 points with the baseline generator. With broader training coverage, the same comparison yields a gain of 0.26 points. The increase of 0.23 points indicates that exposing the generator to more varied transition intervals enables larger gains from adaptive endpoint selection. Under fixed inference, broader coverage slightly improves the four-step score. Its benefit is therefore more apparent when paired with adaptive scheduling. Fig. 4 provides qualitative examples of the four component configurations.

Table 2: **Effects of broader transition coverage and sequential scheduling on VBench full.** All configurations use four sampling steps. Bold marks the best score.

NFE	Broader transition coverage	Sequential scheduling	Evaluation Scores ↑		
			Quality	Semantic	Total
4	✗	✗	85.03	75.42	83.11
	✗	✓	85.02	75.61	83.14
	✓	✗	85.08	75.84	83.23
	✓	✓	**85.28**	**76.33**	**83.49**

Sequential decisions versus initial planning. We examine the role of intermediate-state feedback on VBench medium. An Initial-only planner chooses all endpoints before generation, while Sequential selects each endpoint using the current latent. Their direct comparison also includes differences

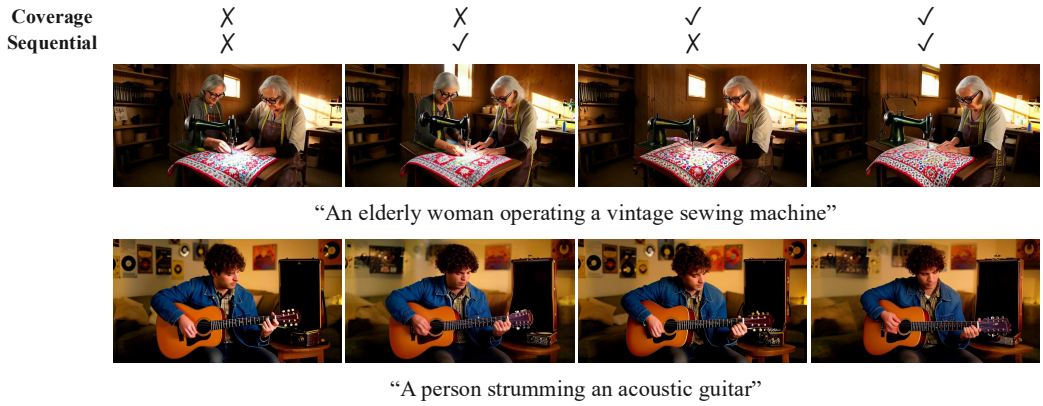

Figure 4: **Qualitative component ablation.** Columns follow the four configurations in Table 2, varying broader transition coverage and sequential scheduling. Each row shows one prompt.

in the first endpoint. We therefore introduce a Shared-first control, which adopts Sequential's first endpoint but plans the remaining endpoints without observing intermediate latents. Shared-first and Sequential execute the same first transition and use the same remaining budget. We report their Total-score difference as feedback gain.

Table 3: **Sequential scheduling versus initial planning on VBench medium.**

NFE	Scheduling policy	Evaluation Scores ↑		
		Quality	Semantic	Total
4	Initial-only	86.12	77.62	84.42
	Shared-first	86.10	78.41	84.56
	Sequential	86.16	78.62	84.65
6	Initial-only	86.38	75.54	84.22
	Shared-first	86.37	75.53	84.20
	Sequential	86.35	76.41	84.36

The results in Table 3 yield feedback gains of 0.0957 and 0.1605 points at four and six steps, respectively. Sequential also exceeds Initial-only by 0.2327 and 0.1448 points. These results show positive gains from state-conditioned continuation relative to the evaluated planner in these settings. The shared-first comparison distinguishes this effect from differences in the initial endpoint choice.

4.4 Efficiency Analysis

End-to-end efficiency. We measure latency on a single NVIDIA A800 using 10 prompts with three repetitions each. Configurations are evaluated in interleaved blocks with warm-up runs. End-to-end timing includes text encoding, sampling, video decoding, and CPU frame output, excluding model loading and video-file encoding and writing. The Total scores in Table 4 come from the full evaluation in Table 1, rather than the timing subset. As shown in Table 4, three-step ReplanFlow takes 9.08 s per video, achieving a 1.21× speedup over four-step AnyFlow while improving the VBench Total score from 83.11 to 83.34. At four steps, ReplanFlow improves the score to 83.49 with essentially unchanged latency. Fig. 5(b) shows the resulting quality–latency tradeoff.

Online scheduling overhead. We isolate scheduling overhead by replaying the same endpoint sequences with the predictor disabled. Online scheduling adds 11.39 ms and 14.96 ms per video at three and four steps, respectively, approximately 0.20% of replay sampling time. This includes state feature processing, prediction, and endpoint selection. All 60 paired runs produce identical final latents. The small scheduling cost explains why ReplanFlow and AnyFlow have nearly iden-

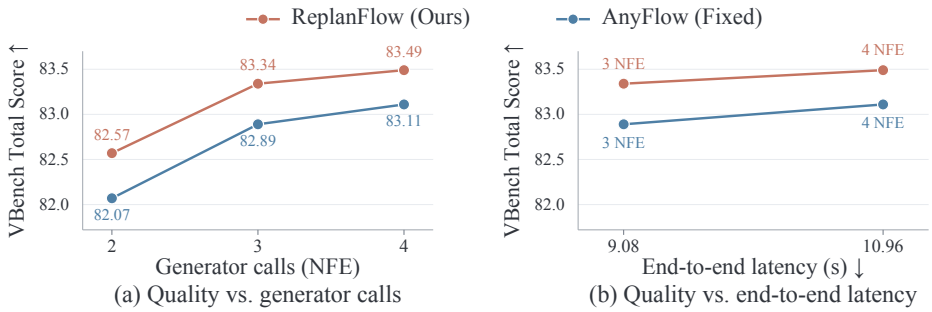

Figure 5: **Quality–efficiency tradeoffs.** VBench full Total scores versus (a) generator calls and (b) end-to-end latency on a single A800. Three-step ReplanFlow exceeds four-step AnyFlow with a 1.21× speedup.

Table 4: **Inference efficiency on a single NVIDIA A800.** Mean latency for 77-frame videos at 448 × 832 resolution and batch size 1. Speedup is relative to four-step AnyFlow.

Model	NFE	Total ↑	Sampling (s) ↓	End-to-end (s) ↓	Speedup ↑	Scheduling overhead (%) ↓
AnyFlow (Fixed)	3	82.89	5.58	9.08	1.21×	—
AnyFlow (Fixed)	4	83.11	7.44	10.96	1.00×	—
ReplanFlow (Ours)	3	83.34	5.59	9.08	1.21×	0.20
ReplanFlow (Ours)	4	**83.49**	7.45	10.96	1.00×	0.20

tical end-to-end latency at the same NFE; variation in preparation and decoding time can mask this additional cost.

5 Conclusion

We presented ReplanFlow, which turns the flexibility of video flow maps into improved generation quality under limited sampling budgets. It combines broader on-policy transition coverage with a lightweight predictor that selects endpoints from the current state and remaining budget. On bidirectional text-to-video generation, ReplanFlow improves VBench Total scores at two, three, and four steps under matched training settings. Its three-step configuration exceeds the four-step AnyFlow baseline while delivering a measured 1.21× end-to-end speedup. The component comparison shows that broader training coverage enables larger gains from adaptive scheduling, with little online overhead.

Limitations. Our predictor relies on offline terminal-quality evaluations tied to a specific generator, adding data-generation cost when adapting to a new model. Learning a shared scheduler across generators and sampling budgets remains an open direction.

AI USE STATEMENT

We used large language models to assist with manuscript drafting and editing, literature search and reference checking, code development, and prompt construction. LLMs were not used as components of the generator or predictor during training or inference, nor as evaluators of experimental outputs. Terminal-quality labels were computed by the evaluation pipeline.

ETHICS STATEMENT

ReplanFlow improves the efficiency of text-to-video generation, which can support creative applications but may also facilitate misleading or harmful synthetic media. Our method inherits the biases and safety limitations of the underlying video generator; improved sampling efficiency does not resolve these issues. We use publicly available training data and describe our data sources and evaluation procedures in the paper. Generated examples are presented as synthetic research outputs and should not be interpreted as recordings of real events.

REPRODUCIBILITY STATEMENT

Section 3 specifies the transition sampling procedure, predictor supervision objective, and budget-constrained inference algorithm. Section 4.1 describes the shared generator-training setup, predictor training, and evaluation protocols. The appendices provide predictor data construction and scoring procedures, architecture and optimization settings, and planning-control implementations. We also report per-dimension VBench scores, training and inference distributions, and qualitative examples with complete prompts and endpoint schedules. Together, these details document the method and the experimental settings used for the reported comparisons.

REFERENCES

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 33346–33382, 2025a. doi: 10.52202/085713-1121.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow Map Matching with Stochastic Interpolants: A Mathematical Framework for Consistency Models. *Transactions on Machine Learning Research*, 2025b.

Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the Trajectory Regularity of ODE-based Diffusion Sampling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7905–7934. PMLR, 2024.

Xin Ding, Lei Yu, Xin Li, Zhijun Tu, Hanting Chen, Jie Hu, and Zhibo Chen. RaSS: Improving Denoising Diffusion Samplers with Reinforced Active Sampling Scheduler. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12923–12933, 2025.

Eric Frankel, Sitan Chen, Jerry Li, Pang Wei Koh, Lillian J. Ratliff, and Sewoong Oh. S4S: Solving for a Fast Diffusion Model Solver. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 17542–17574. PMLR, 2025.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One Step Diffusion via Shortcut Models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Xingtong Ge, Yi Zhang, Yushi Huang, Dailan He, Xiahong Wang, Bingqi Ma, Guanglu Song, Yu Liu, and Jun Zhang. SALT: Self-Consistent Distribution Matching with Cache-Aware Training for Few-Step Video Generation. In *European Conference on Computer Vision*, 2026.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean Flows for One-step Generative Modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 75460–75482, 2025. doi: 10.52202/085713-2534.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved Mean Flows: On the Challenges of Fastforward Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30467–30476, 2026.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. AnyFlow: Any-Step Video Diffusion Model with On-Policy Flow Map Distillation. In *European Conference on Computer Vision*, 2026.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 167283–167308, 2025. doi: 10.52202/085713-5576.

Yilie Huang, Wenpin Tang, and Xunyu Zhou. ART for Diffusion Sampling: A Reinforcement Learning Approach to Timestep Schedule. *arXiv preprint arXiv:2601.18681*, 2026.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive Benchmark Suite for Video Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024.

Binglin Ji, Anindya Sarkar, Hengchang Lu, Jens Sjölund, and Yevgeniy Vorobeychik. Bootstrap Flow-Map Tree Sampling Enables Online Feedback Driven Search. *arXiv preprint arXiv:2607.02915*, 2026.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion. In *International Conference on Learning Representations*, volume 2024, pp. 44493–44525, 2024.

Thanh-Tung Le, Yunhan Zhao, Menglei Chai, Zhengyang Shen, Zhe Cao, Danhang Tang, Xiaohui Xie, and Deying Kong. DSA: Dynamic Step Allocation for Fast Autoregressive Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 4334–4344, 2026.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion Adversarial Post-Training for One-Step Video Generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37959–37974. PMLR, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023.

Tao Liu, Hao Yan, Mengting Chen, Taihang Hu, Zhengrong Yue, Zihao Pan, Jinsong Lan, Xiaoyong Zhu, Ming-Ming Cheng, Bo Zheng, and Yaxing Wang. Continuous-Time Distribution Matching for Few-Step Diffusion Distillation. *arXiv preprint arXiv:2605.06376*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023.

Cheng Lu and Yang Song. Simplifying, Stabilizing and Scaling Continuous-Time Consistency Models. In *International Conference on Learning Representations*, volume 2025, pp. 50611–50649, 2025.

Bao Nguyen, Binh Nguyen, and Viet Anh Nguyen. Bellman Optimal Stepsize Straightening of Flow-Matching Models. In *International Conference on Learning Representations*, volume 2024, pp. 30948–30968, 2024.

Weili Nie, Julius Berner, Nanye Ma, Chao Liu, Saining Xie, and Arash Vahdat. Transition Matching Distillation for Fast Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4645–4655, 2026.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align Your Steps: Optimizing Sampling Schedules in Diffusion Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42947–42975. PMLR, 2024.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align Your Flow: Scaling Continuous-Time Flow Map Distillation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 146459–146512, 2025. doi: 10.52202/085713-4900.

Neta Shaul, Chao Liu, Arash Vahdat, and Julius Berner. Parallel Decoding Distillation for Fast Image and Video Generation. *arXiv preprint arXiv:2607.26004*, 2026.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023.

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.

Vinh Tong, Trung-Dung Hoang, Anji Liu, Guy Van den Broeck, and Mathias Niepert. Learning to Discretize Denoising Diffusion ODEs. In *International Conference on Learning Representations*, volume 2025, pp. 47244–47282, 2025.

Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating Diffusion Sampling with Optimized Time Steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8292–8301, 2024.

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*, volume 2025, pp. 83048–83077, 2025.

Zilyu Ye, Zhiyang Chen, Tiancheng Li, Zemin Huang, Weijian Luo, and Guo-Jun Qi. Schedule On the Fly: Diffusion Time Prediction for Faster and Better Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23412–23422, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487, 2024a. doi: 10.52202/079017-1505.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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 (CVPR)*, pp. 6613–6623, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From Slow Bidirectional to Fast Autoregressive Video Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, 2025.

Peiyu Yu, Suraj Kothawade, Sirui Xie, Ying Nian Wu, and Hongliang Fei. Designing Instance-Level Sampling Schedules via REINFORCE with James–Stein Shrinkage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36040–36050, 2026.

Liangyu Yuan, Ruoyu Wang, Tong Zhao, Dingwen Fu, Mingkun Lei, Beier Zhu, and Chi Zhang. Few-Step Diffusion Sampling Through Instance-Aware Discretizations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35882–35892, 2026.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large Scale Diffusion Distillation via Score-Regularized Continuous-Time Consistency. In *International Conference on Learning Representations*, volume 2026, pp. 2582–2603, 2026.

Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ODE-based Sampling for Diffusion Models in Around 5 Steps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7777–7786, 2024.

Aihua Zhu, Rui Su, Qinglin Zhao, Li Feng, Meng Shen, and Shibo He. Hierarchical Schedule Optimization for Fast and Robust Diffusion Model Sampling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16):13907–13915, 2026. doi: 10.1609/aaai.v40i16.38400.

APPENDIX CONTENTS

A Training Coverage and Inference Endpoint Distributions 15

B Per-Dimension VBench Results 16

C Predictor Supervision Construction 18

D Predictor Architecture and Optimization 18

E Additional Qualitative Examples 20

A Training Coverage and Inference Endpoint Distributions

Inference endpoint selection. Fig. 6(a) shows the internal endpoints selected by the three- and four-step predictors. The distributions vary across trajectories and differ from the fixed-schedule references. First endpoints tend to concentrate at higher noise levels, while later endpoints occupy several distinct regions. The learned predictors therefore use sample-dependent schedules rather than reproducing a single preset schedule.

On-policy transition coverage. Fig. 6(b) compares the selected Stage 2 intervals under matched sampled schedule lengths and interval positions. AnyFlow's preset schedules produce a sparse set of time pairs, whereas nonuniform sampling spreads the observed transitions over a broader region. This exposes the generator to more varied finite-time transitions during distillation. Together with the component ablation in Table 2, these observations support broadening training coverage to better support adaptive endpoint selection.

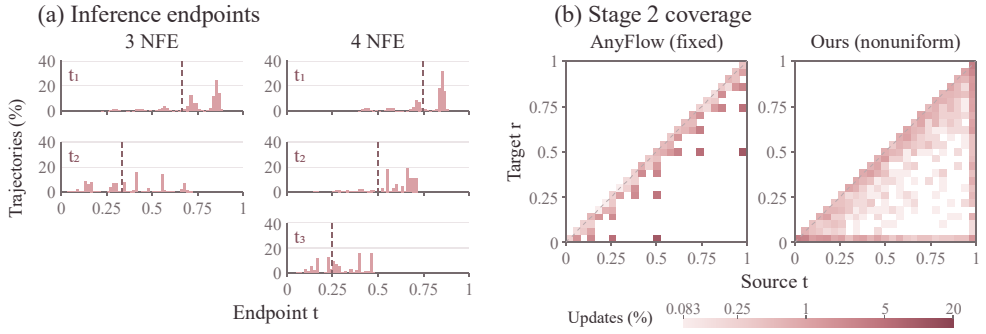

Figure 6: **Training transition coverage and inference endpoint distributions.** All times use normalized pre-shift coordinates. **(a)** Internal endpoints from 4,720 trajectories per budget; bars show trajectory percentages in bins of width $1/64$, and dashed lines mark fixed-schedule endpoints. The forced endpoint at zero is omitted. **(b)** Selected distillation intervals from 1,200 generator updates per model, shown in 25×25 bins with a shared logarithmic color scale. Shortcut prefix and suffix transitions are excluded.

B PER-DIMENSION VBENCH RESULTS

We report all 16 dimensions for the controlled main comparisons in Table 5 and the component ablation in Table 6. Both tables use VBench full. Dimension scores are shown before the normalization used to compute the aggregate scores, multiplied by 100. Quality, Semantic, and Total retain the benchmark aggregation used in the main tables; they are not simple averages of the displayed dimension scores. The AnyFlow rows refer to our controlled reproduction. Published reference models in Table 1 are not re-evaluated here.

Table 5: **Per-dimension VBench full scores for the controlled comparisons in Table 1.**

	AnyFlow (Fixed)			**ReplanFlow (Sequential)**		
Dimension	**2 NFE**	**3 NFE**	**4 NFE**	**2 NFE**	**3 NFE**	**4 NFE**
Quality dimensions						
Subject consistency	95.19	95.83	95.82	95.38	95.89	96.02
Background consistency	96.33	96.40	96.29	96.38	96.39	96.49
Temporal flickering	99.28	99.37	99.40	99.23	99.31	99.36
Motion smoothness	98.45	98.46	98.43	98.49	98.48	98.48
Dynamic degree	62.50	63.06	64.44	62.50	65.83	65.00
Aesthetic quality	67.65	69.08	69.15	67.72	69.09	69.50
Imaging quality	64.61	66.60	67.35	65.27	67.34	67.84
Semantic dimensions						
Object class	86.46	88.09	90.14	89.21	90.35	89.56
Multiple objects	75.87	81.01	80.20	79.62	80.61	81.89
Human action	92.00	92.60	93.80	93.00	92.60	94.20
Color	88.25	89.40	89.87	91.34	91.78	91.86
Spatial relationship	74.32	75.33	74.53	73.82	75.76	75.04
Scene	38.74	39.99	39.36	42.31	41.37	41.77
Appearance style	20.73	20.53	20.58	21.16	20.78	20.74
Temporal style	22.08	22.24	22.39	22.20	22.47	22.53
Overall consistency	25.06	25.04	25.07	25.29	25.26	25.15
Quality	84.08	84.82	85.03	84.24	85.14	85.28
Semantic	74.01	75.20	75.42	75.88	76.14	76.33
Total	82.07	82.89	83.11	82.57	83.34	83.49

Table 6: **Per-dimension VBench full scores for the four-step component ablation in Table 2.**

Broader coverage	✗	✗	✓	✓
Sequential scheduling	✗	✓	✗	✓
Quality dimensions				
Subject consistency	95.82	95.85	95.94	96.02
Background consistency	96.29	96.41	96.47	96.49
Temporal flickering	99.40	99.36	99.39	99.36
Motion smoothness	98.43	98.45	98.44	98.48
Dynamic degree	64.44	64.44	63.61	65.00
Aesthetic quality	69.15	69.08	69.37	69.50
Imaging quality	67.35	67.22	67.52	67.84
Semantic dimensions				
Object class	90.14	89.94	89.30	89.56
Multiple objects	80.20	80.12	80.66	81.89
Human action	93.80	93.60	93.80	94.20
Color	89.87	89.39	92.24	91.86
Spatial relationship	74.53	74.86	75.12	75.04
Scene	39.36	40.42	40.41	41.77
Appearance style	20.58	20.72	20.59	20.74
Temporal style	22.39	22.52	22.36	22.53
Overall consistency	25.07	25.13	25.06	25.15
Quality	85.03	85.02	85.08	85.28
Semantic	75.42	75.61	75.84	76.33
Total	83.11	83.14	83.23	83.49

C Predictor Supervision Construction

Prompts and candidate continuations. We use approximately 1K GPT-generated prompts covering diverse scene categories and semantic attributes, with five generation seeds per prompt and the same prompt set across budgets. For each prompt–seed pair, we enumerate decreasing schedules on an eight-point pre-shift grid from 1 to 0. Selecting $B - 1$ internal endpoints gives 6, 15, and 20 schedules for two-, three-, and four-step generation. We reuse shared prefixes and retain states with at least two candidate actions. Each action receives the highest terminal-quality score among its enumerated continuations, as defined in Eq. 5.

Terminal-quality supervision. The training reward combines normalized VBench quality dimensions and applicable semantic dimensions with weights 0.8 and 0.2. This per-video score differs from the benchmark-level VBench Total score.

Dataset size and offline cost. Across all prompts and seeds, the two-, three-, and four-step datasets contain approximately 28.3K, 89.7K, and 136.9K state–action labels, respectively. Constructing the complete four-step dataset, including candidate generation and scoring across all five seeds, requires approximately 60 A800 GPU-hours.

D Predictor Architecture and Optimization

State encoding and fusion. Fig. 7 shows the four input branches; Table 7 gives their dimensions. Latents use fixed per-channel normalization statistics from the corresponding budget's seed-0 training data. The history encoder receives Fourier features of endpoints and gaps, using a learned vector at the root. The time/budget encoder combines Fourier features of t_k, a budget embedding, and $(1/B_k, t_k/B_k, k)$. Each branch produces one token. A learned policy token and token-type embeddings complete the five-token input to pre-normalized attention blocks. The categorical head reads the final policy token; endpoint selection follows Section 3.4.

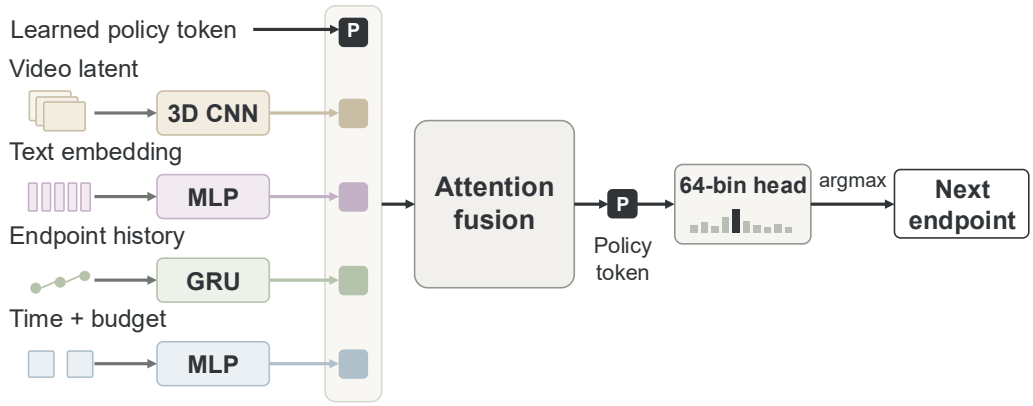

Figure 7: **Endpoint predictor architecture.** Four state encoders and a learned policy token feed a compact attention module. The categorical head selects relative progress, which is converted to the next endpoint.

Table 7: **Predictor architecture.** Kernel and stride tuples follow temporal, height, and width axes.

Component	Configuration
Latent stem	Conv3D, $16 \to 48$ channels; kernel/stride $(1, 4, 4)$
Residual blocks	Output channels 64, 96, 128, 128; strides $(2, 2, 2)$, $(2, 1, 1)$, $(1, 1, 1)$, $(1, 1, 1)$; depthwise kernel $(3, 3, 3)$
Latent pooling	Two learned 128-dimensional queries; 256-dimensional output
Text encoder	Mean-pool valid text tokens; MLP $4096 \to 64 \to 128$
History encoder	34-dimensional Fourier features; GRU hidden width 64
Time/budget encoder	17 time features, 16-dimensional budget embedding, 3 scalars; MLP $36 \to 64 \to 64$
Token projections	Latent/text/history/time-budget features to width 128
Attention fusion	2 blocks, 4 heads, width 128; feed-forward width 512
Output head	Linear $128 \to 64$ relative-progress logits

Optimization and curriculum. For the main results, each budget has a separate predictor, trained from random initialization on its full constructed dataset with the generator frozen. All predictor modules are optimized jointly using Eq. 7. Table 8 centralizes the training settings. The curriculum expands from root states to depths ≤ 1, then to all available depths. Two-step data contain only depth 0; three-step data contain depths 0 and 1.

Table 8: **Training settings for the main-result predictors.**

Setting	Value
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Training updates / batch size	5,000 / 48 states
Initial learning rate (all modules)	5×10^{-5}
Learning-rate schedule	Cosine decay to 10^{-6} over 5,000 updates
Weight decay / gradient clipping norm	0 / 5
Attention and feed-forward dropout	0
Target temperature τ / bandwidth σ	0.002 / 0.06
Curriculum: root / depths ≤ 1 / all depths	250 / 500 / 4,250 updates
Minibatch sampling	Balanced across active decision depths
Initialization seed	20260903

Planning controls on VBench medium. Unlike the main experiments, Table 3 uses seed-0 supervision and 192 evaluation prompts with generation seed 0. Matched Sequential and Initial-only predictors use 1,500 updates at four NFE and the selected 3,000-update curriculum configuration at six NFE. Initial-only retains the architecture and training recipe but replaces each current-latent input with the corresponding initial noise. It autoregressively plans all endpoints before generation, updating time, budget, and endpoint history without observing intermediate latents. Shared-first uses Sequential's first endpoint, then commits to an Initial-only suffix before executing it. The shared first transition produces the same latent as Sequential; only Sequential subsequently observes updated latents.

E ADDITIONAL QUALITATIVE EXAMPLES

We present a broader collection of qualitative examples with complete input prompts, including the examples shown in the main text. Comparisons include four-step AnyFlow (Fixed) versus three- and four-step ReplanFlow (Sequential), as well as four-step comparisons among AnyFlow (Fixed), AnyFlow (Sequential), and ReplanFlow (Sequential). Each figure shows sampled frames and end-point schedules, with configurations specified by the row labels.

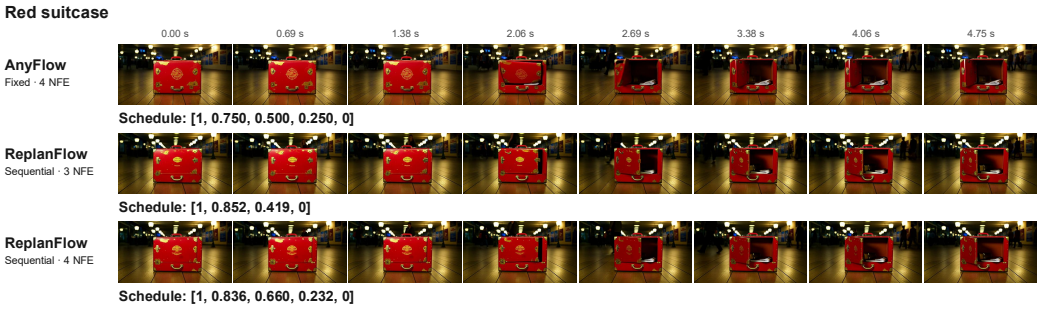

Figure 8: **Red suitcase. Prompt:** A vintage red suitcase stands prominently in the center of a dimly lit airport terminal. The suitcase is adorned with faded golden lettering and intricate patterns. It sits on a worn wooden floor, catching the soft glow of several nearby street lamps. The suitcase has a slightly battered look, with peeling paint and scratch marks on the sides. A single suitcase handle is slightly raised, indicating it was recently opened. The suitcase door is partially open, revealing a few items inside, including a worn leather wallet and a stack of crumpled papers. In the background, a group of travelers walks hurriedly past, their faces blurred against the dim lighting. The terminal's walls are lined with posters of famous destinations, casting intriguing shadows across the scene. Soft ambient airport sounds fill the air, adding to the nostalgic atmosphere. The suitcase emits a gentle hum as if communicating silently, inviting viewers to explore its secrets. Medium shot, slightly low-angle view.

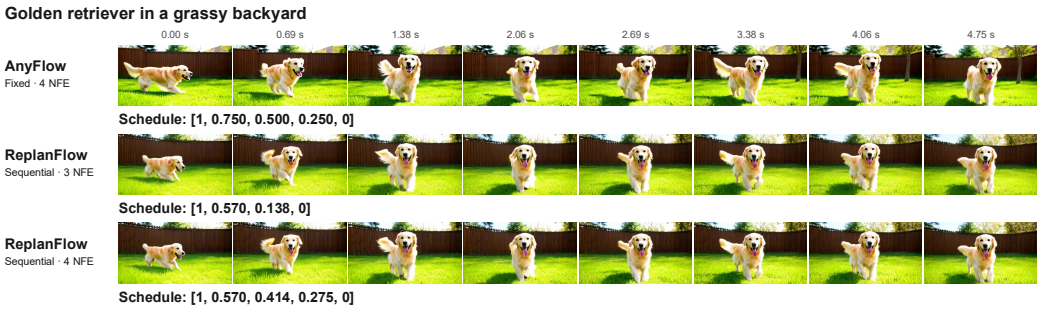

Figure 9: **Golden retriever in a grassy backyard. Prompt:** a golden retriever playing in a grassy backyard. The dog runs back and forth energetically, occasionally stopping to look at the camera with its tongue out and tail wagging. The background includes a wooden fence and a few trees, suggesting a suburban setting. The lighting is bright and natural, indicating that the video was shot on a sunny day. The style is cheerful and lighthearted, focusing on the dog's playful behavior.

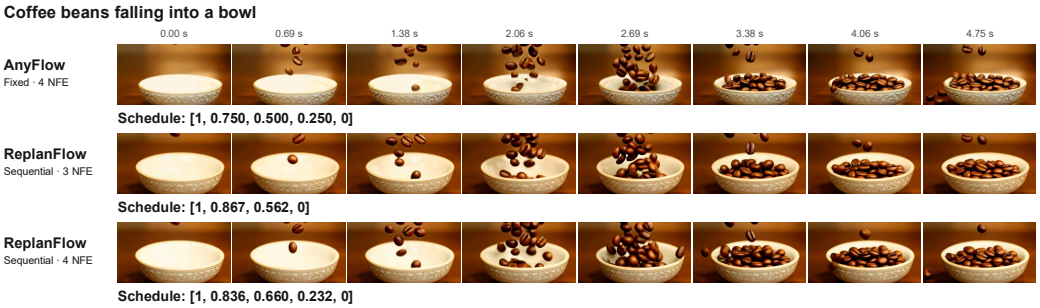

Figure 10: **Coffee beans falling into a bowl. Prompt:** Macro slo-mo. Slow motion cropped closeup of freshly roasted coffee beans cascading gracefully into an empty porcelain bowl. The coffee beans, golden brown and glossy, gently land one by one, each with a soft crunch as they hit the bowl's surface. The bowl, made of fine ceramic with intricate patterns etched into its sides, reflects the warm, ambient lighting from the kitchen. The scene is bathed in a gentle, diffused light, casting a warm glow over the entire setup. The camera moves slightly from side to side, capturing the subtle textures and colors of each bean as it descends. The air is filled with the rich, aromatic scent of roasting coffee beans. The bowl gradually fills up, creating a mesmerizing display of nature's slow-motion ballet. Macro shot focusing on the individual beans, medium shot of the bowl, close-up of the beans just before they enter the bowl.



Figure 11: **Blue umbrella. Prompt:** A blue umbrella standing outdoors on a sunny day. The umbrella is held open by a young woman in a flowy floral dress, with a vibrant smile on her face. She is walking towards a quaint café, her hair blowing gently in the breeze. The background features a lush green park with tall trees and colorful flowers. Soft natural lighting highlights the woman's expressive eyes and the intricate details of the umbrella. The scene captures a moment of casual elegance and joy. Medium shot, handheld camera movement capturing the woman and the umbrella.

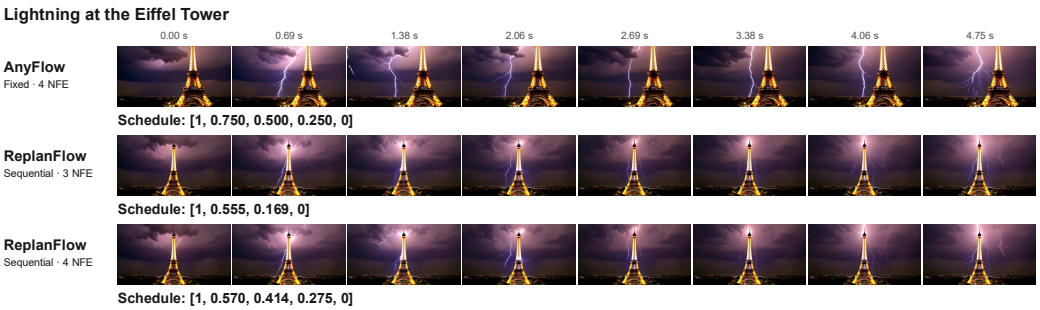

Figure 12: **Lightning at the Eiffel Tower. Prompt:** A dramatic lightning strike illuminating the iconic Eiffel Tower against a backdrop of dark, ominous clouds in the sky. The lightning crackles and illuminates the structure, casting stark shadows and highlighting every intricate detail. Dark storm clouds swirl menacingly overhead, adding to the intense atmosphere. The Eiffel Tower stands tall and proud amidst the storm, its metal lattice frame gleaming in the electric flash. The scene captures the raw power and beauty of nature's fury, with the lightning slicing through the sky. The lightning bolt strikes the tower, sending a shockwave through the air as it leaves a trail of sparks. The shot includes the lightning bolt from various angles, showcasing its intensity and impact. The background features a chaotic mix of dark clouds, raindrops, and blurred city lights. The lightning bolt arcs across the sky, creating a mesmerizing display. The lighting is dramatic and contrasts sharply with the dark, stormy clouds. The Eiffel Tower is captured in a medium shot, emphasizing its grandeur and vulnerability in the face of nature's wrath.



Figure 13: **Vintage sewing machine. Prompt:** Vintage sewing machine in a cluttered workshop, an elderly woman with silver hair and gentle eyes carefully operating the old machine. She wears a faded apron and a pair of thick-rimmed glasses. The machine hums softly as she stitches intricate patterns into a colorful quilt. The workshop is dimly lit, with dusty shelves lined with various tools and old fabric samples. She occasionally pauses to adjust the tension and thread, her hands moving deftly. Soft sunlight filters through the window, casting warm shadows across the wooden floorboards. The background features worn wooden walls and cobweb-covered corners. Antique sewing machine texture with a vintage feel. Close-up of the woman's hands at the sewing machine, medium shot of the woman and the sewing machine, and wide shot of the workshop.

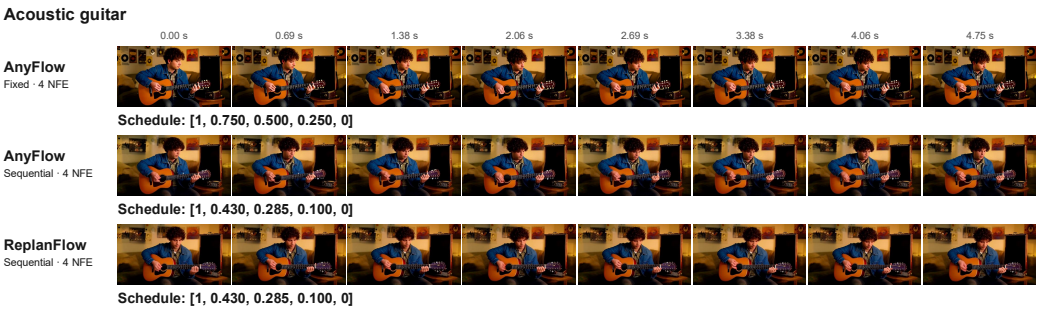

Figure 14: **Acoustic guitar. Prompt:** A skilled musician strums a classic acoustic guitar with intricate fretwork, their fingers moving gracefully across the strings. The guitarist has tousled brown hair and expressive eyes, dressed in a vintage denim jacket and flannel shirt. They sit at a small wooden table in a cozy, dimly lit living room, surrounded by old vinyl records and colorful posters. Soft ambient lighting casts warm shadows, enhancing the intimate atmosphere. The guitar case stands open beside them, filled with various instruments and musical accessories. The background features worn-out couches and scattered cushions, adding to the lived-in feel. The scene captures a moment of focused concentration and passion as the guitarist's fingers dance over the strings, creating a beautiful melody. Cinematic close-up shot focusing on the musician's face and hands.

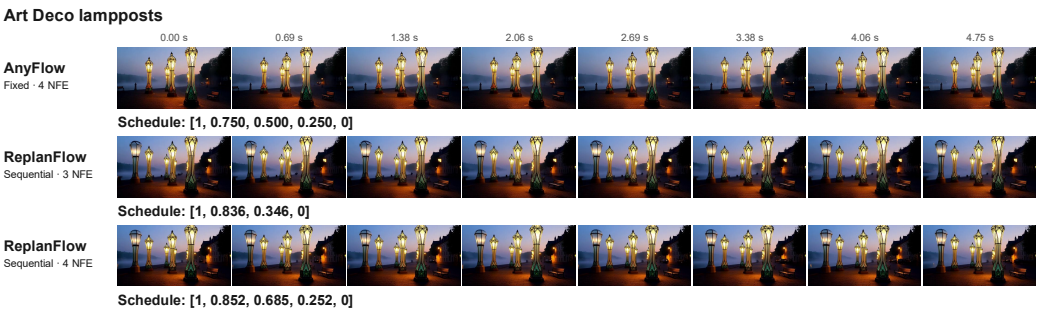

Figure 15: **Art Deco lampposts. Prompt:** A tranquil tableau of the lampposts adorned with Art Deco motifs, their geometric shapes and frosted glass create a sense of vintage glamour. The scene features a row of ornate lampposts lining a quiet street at dusk. Each lamppost stands tall, casting soft shadows and golden hues across the cobblestone pavement. The lampposts are intricately designed with flowing lines and bold geometric patterns, their surfaces covered in frosted glass panels that refract light into mesmerizing prismatic colors. A single, elegant bench sits nearby, inviting passersby to take a seat and savor the moment. The backdrop is a serene, twilight landscape with a few wisps of mist rising from the river in the distance. The atmosphere is calm and nostalgic, filled with the gentle hum of street life and the distant sound of carriages passing by. The scene captures the essence of a bygone era, blending modern elegance with timeless charm. Soft, ambient lighting enhances the ambiance, casting a warm glow over the entire tableau. Medium shot, focusing on the detail-rich lampposts and the tranquil setting.



Figure 16: **Skateboarding through the city. Prompt:** A person is skateboarding down a winding urban street at dusk. The skateboarder wears a black hoodie with reflective stripes and cargo pants, his long blonde hair blowing in the evening breeze. He glides smoothly over cobblestone and graffiti-covered surfaces, occasionally jumping over small obstacles. The street lights flicker on as he maneuvers between parked cars and narrow alleyways. The background features towering skyscrapers and neon signs, casting dramatic shadows. The scene captures the thrill and energy of urban skateboarding. Low-angle shot focusing on the skateboarder's face, showcasing his focused expression and dynamic movement.

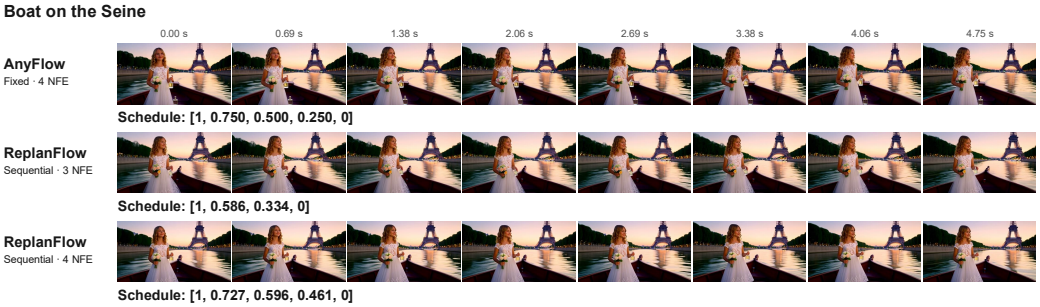

Figure 17: **Boat on the Seine. Prompt:** A picturesque scene captured in fresh film photography, a young French girl with wavy blonde hair and wearing a flowing white dress with intricate lace patterns. She is holding a small bouquet of flowers in one hand and a bottle of perfume in the other, as she stands gracefully by the bow of a classic wooden boat. The boat is gently gliding along the serene Seine River, with the iconic Eiffel Tower towering majestically in the twilight sky behind them. The girl gazes out at the water, a serene smile playing on her lips. The background is a soft, gradient sunset with hints of orange and purple hues, blending seamlessly into the night. The boat tilts downward, capturing the gentle sway of the water and the reflection of the city lights on the river surface. A vintage film texture photo, medium shot from the side, focusing on the girl and the boat.

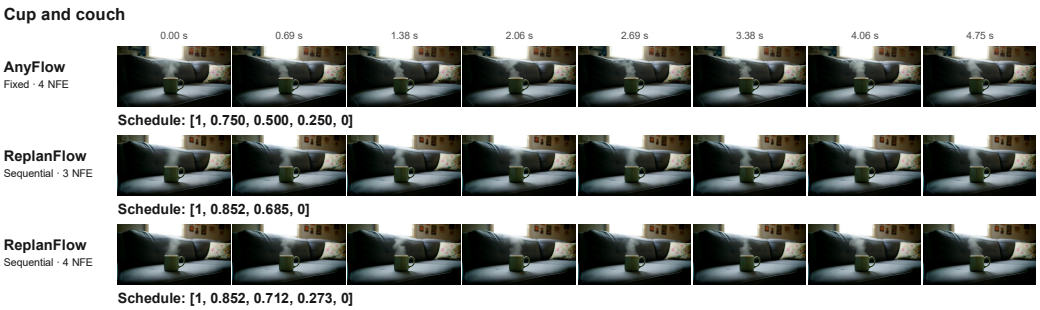

Figure 18: **Cup and couch. Prompt:** A cozy scene captured in soft lighting, a vintage ceramic mug filled with steaming coffee sits on a plush gray couch in a small living room. The mug's handle is slightly warm to the touch, and steam rises from the surface. The couch is worn but comfortable, with tufted pillows and faded floral patterns. The room is dimly lit, with gentle shadows accentuating the textures of the furniture. A single window allows in a soft glow, casting dappled light across the space. The walls are adorned with eclectic art pieces and vintage posters, adding a touch of personality to the setting. The overall atmosphere is inviting and intimate, inviting viewers to feel at home. The scene captures a moment of quiet reflection, with the viewer's gaze drawn to the mug and the couch as if in a state of contentment. Soft ambient sounds of the room fill the air, creating a serene ambiance. Medium shot, side view.

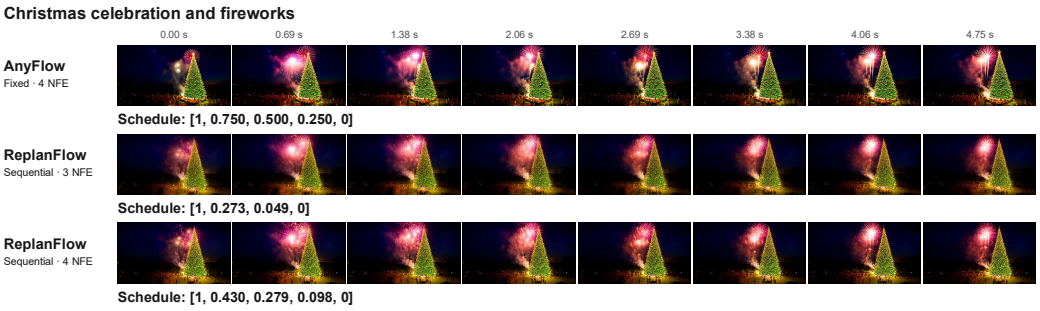

Figure 19: **Christmas celebration and fireworks. Prompt:** A drone view capturing a festive celebration, featuring a towering Christmas tree adorned with twinkling lights and colorful ornaments. The sky is filled with vibrant fireworks, creating a spectacular display against a backdrop of a starry night. The scene is illuminated by the soft glow of the fireworks, blending seamlessly with the twinkling stars. In the foreground, revelers are seen dancing and enjoying the festivities, their faces aglow with joy. The camera pans across the expansive landscape, showcasing the beauty of the celebration from above. The overall atmosphere is warm and welcoming, with a sense of community and excitement. Cinematic drone footage with sweeping aerial shots and dynamic camera movements.



Figure 20: **Person and sink. Prompt:** CG game concept digital art, a person standing next to a large, ornate bathroom sink in a dimly lit room. The person has long, wavy brown hair and piercing blue eyes, wearing a stylish leather jacket over a tight-fitting black t-shirt. They lean against the sink, one hand resting on the edge, the other holding a partially opened water bottle. The bathroom is filled with vintage fixtures, including claw-foot bathtubs and chandeliers. The walls are adorned with faded wallpaper featuring floral patterns. Soft ambient lighting casts a warm glow, creating a cozy atmosphere. The person appears lost in thought, their posture slightly hunched as they gaze into the mirror hanging above the sink. The background showcases intricate tile work and a vintage medicine cabinet. Darker tones and muted colors give the scene a nostalgic feel. Low-angle, close-up shot focusing on the person's reflection in the sink mirror.