SPEND WHERE IT MATTERS: PARTICLE SEARCH WITH MH REFINEMENT FOR BUDGET-CONSTRAINED AUDIO-VISUAL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward-guided inference-time alignment can improve joint audio-visual generation, but optimizing multiple alignment objectives requires many expensive model evaluations, making existing methods costly under limited compute. Particle-based search enables global exploration but incurs estimation error with a finite population, while Metropolis-Hastings (MH) can refine candidates locally but benefits from a strong initialization. We introduce **S**pend **W**here **I**t **M**atters (SWIM), a two-stage, training-free, reward-agnostic framework that combines particle search with MH refinement under a fixed compute budget. In the first stage, we adaptively allocate particles across denoising checkpoints using online variance estimates, minimizing estimation variance while concentrating compute where it is most useful. This search provides a promising candidate for MH refinement, which uses local noise-and-denoise proposals on completed outputs to further improve reward. By starting from a strong candidate, MH can recover additional reward without requiring a larger particle population or costly global exploration. We show that, at matched compute, combining particle search with MH refinement achieves higher rewards than either approach alone. On JAVISBENCH-MINI, at closely matched compute, SWIM achieves mean relative gains across seven metrics of 30.7% over BoN and 14.1% over EvoSearch with JavisDiT++, and 9.2% over BoN and 17.0% over EvoSearch with LTX-2. Beyond joint audio-visual generation, we also demonstrate the benefits of adaptive particle allocation on SD1.5 image generation, improving AestheticScore by 2.2% and ImageReward by 6.0% over uniform allocation.

1 INTRODUCTION

Diffusion and flow-based models have achieved high-fidelity generation across images (Rombach et al., 2022; Esser et al., 2024), video (Blattmann et al., 2023; Wan et al., 2025), text (Nie et al., 2025; Sahoo et al., 2024), and audio (Liu et al., 2023; Evans et al., 2024). As individual modalities mature, attention has shifted toward *joint multi-modal generation*. Joint audio-visual (AV) video generation is particularly compelling because audio and visual signals are closely coupled in the real world, with direct applications in media such as film, gaming, and AR/VR. While recent models such as JavisDiT++ (Liu et al., 2026b) and LTX-2 (HaCohen et al., 2026) jointly generate visuals and audio from text within a single flow-matching framework, they still struggle to align the generation with the text prompts, as well as align the generated audio and visual modalities with each other.

Such alignment issues are well-studied in text and image generation, and two main approaches have emerged. *Post-training alignment* uses RL to align generative models with rewards (Black et al., 2024; Shekhar et al., 2025; Shekhar & Zhang, 2025; Clark et al., 2024), and *inference-time alignment* steers generation toward higher-reward outputs without modifying model weights (Singhal et al., 2025; Shekhar et al., 2026; Jain et al., 2025). In this work, we adopt inference-time alignment for joint AV generation because it is training-free and modular, allowing us to adapt to different alignment objectives simply by swapping reward models without retraining the generator.

Among inference-time alignment methods, *particle-based methods* have shown promise as they offer efficient, parallelizable global exploration without requiring reward gradients (Singhal et al.,

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.

2025; Skreta et al., 2025; Shekhar et al., 2026). However, limited particle populations can lead to high estimation variance and degraded sample quality. Metropolis-Hastings (MH) (Lee et al., 2025) enables local refinement but requires substantial compute for global exploration due to slow mixing (Roberts et al., 1997). More recently, ITS-AVGen (Jung et al., 2026) explores joint AV alignment using multiple verifiers and adaptive reward weighting to optimize coarse and fine-grained objectives simultaneously. However, fine-grained reward estimates can be unreliable before the coarse AV structure is established, potentially degrading alignment.

To address these challenges, we introduce **S**pend **W**here **I**t **M**atters (SWIM), a two-stage inference-time alignment framework that combines budget-constrained particle search with MH refinement for coarse-to-fine alignment. Motivated by VASR (Shekhar et al., 2026), we formulate particle allocation as a variance-minimization problem under a fixed compute budget and derive a closed-form optimal schedule. Using online sampling-variance estimates, we adapt the particle population across denoising checkpoints to establish coarse alignment and obtain a promising candidate under a fixed budget. MH then refines this candidate through local noise-and-denoise proposals, optimizing fine-grained alignment after the coarse structure is established. This division of labor reduces the need for costly global exploration during MH while recovering reward without requiring a larger particle population, enabling efficient multi-objective alignment under a fixed compute budget.

Our contributions are:

- **Particle-initialized MH for budget-constrained alignment.** We introduce a two-stage strategy that first explores several denoising trajectories with particle search, then uses the particle with the highest particle-stage reward to initialize a short MH chain. For the same total number of model evaluations, this strategy achieves a higher mean rewards than particle search alone or MH from an unguided sample at every tested budget.
- **Online variance-based particle allocation.** We formulate particle allocation across denoising checkpoints as a variance-minimization problem under a fixed budget and derive a closed-form solution. During each generation, we use online variance estimates to adapt the particle population at each checkpoint to the prompt and the corresponding reward signals.
- **Evaluation on joint audio-visual generation.** On JAVISBENCH-MINI, at closely matched compute, SWIM achieves mean relative gains across seven metrics of 30.7% over BoN and 14.1% over EvoSearch with JavisDiT++, and 9.2% over BoN and 17.0% over EvoSearch with LTX-2. In image generation, on SD1.5 at 400 forward passes, it improves AestheticScore by 2.2% and ImageReward by 6.0% over uniform allocation.

2 RELATED WORK

Audio-visual alignment. Recent work has explored both training-based and inference-time approaches to improving alignment in joint audio-visual video generation. SyncDPO (Cheng et al., 2026) applies direct preference optimization using temporally misaligned negatives, while SpongeBob (Liang et al., 2026) introduces a dual-stream Diffusion Transformer with bidirectional cross-modal attention and synchronization-aware training. These methods improve alignment by modifying the underlying generative model, whereas our method operates entirely at inference time and can be applied to pretrained backbones without additional training. More recently, ITS-AVGen (Jung et al., 2026) studies inference-time scaling for joint AV generation using multiple verifiers and introduces Adaptive Reward Weighting (ARW) to calibrate heterogeneous reward signals at test time. In contrast, our method performs multi-stage inference-time alignment, first achieving coarse alignment and then applying a finer adjustment stage for further refinement, without additional training.

Efficient inference-time alignment. Our approach builds on inference-time alignment and particle-based sampling for diffusion models. Best-of-N sampling generates independent candidates but cannot reuse computation from unsuccessful trajectories. Feynman-Kac (FK) steering (Singhal et al., 2025) instead uses interacting particles and intermediate resampling guided by reward-derived potentials, while FK Correctors (Skreta et al., 2025) extend this framework to multi-objective generation through product-of-experts distributions. VASR (Shekhar et al., 2026) reduces SMC variance through variance-optimal particle resampling, whereas DTS* (Jain et al., 2025) uses a monte-carlo tree search to find high reward sample. Particle-based search offers global exploration and can reuse computation, but finite particle populations introduce estimation error when approximating

the reward-twisted distribution. Building on VASR, we further reduce estimation variance by adaptively allocating particles across denoising checkpoints under a fixed compute budget. Our allocation strategy is compatible with different resampling rules. We refine the candidate using MH, which achieves greater reward improvements than allocating the same compute to additional particles.

3 PRELIMINARIES

Diffusion and flow-matching models. Denoising diffusion probabilistic models (Ho et al., 2020) use the forward process $q(x_t \mid x_0) = \mathcal{N}(x_t;\, \sqrt{\bar{\alpha}_t}\, x_0,\, (1-\bar{\alpha}_t)I)$ and train a noise predictor $\epsilon_\theta(x_t, t)$. In a continuous-time score formulation, the reverse-time SDE and probability-flow ODE share marginals but have different drifts (Song et al., 2021). Flow matching (Lipman et al., 2023; Liu et al., 2022) trains a velocity field $v_\theta(x_t, t) \approx \varepsilon - x_0$ on the interpolant $x_t = (1-t)x_0 + t\varepsilon, \varepsilon \sim \mathcal{N}(0, I_d)$, and integrating backward from noise gives a noise-to-data flow. We write our sampler as

$$dX_t = \underbrace{\tilde{b}_\theta(X_t, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{diffusion}}\, dW_t, \tag{1}$$

where drift and diffusion depend on the sampler. Our flow-model sampler uses $g(t) = \eta\, t$. The denoised estimate $\hat{x}_0(x_t, t)$ approximates $\mathbb{E}[x_0 \mid x_t]$ and can be decoded for reward evaluation. We use the sampler associated with each underlying model.

Sequential Monte Carlo (SMC). For a reference generator $p_{\text{ref}}(x \mid p)$ and reward $R(x, p)$, the ideal reward-tilted target is $\pi(x \mid p) \propto p_{\text{ref}}(x \mid p) \exp(\lambda R(x, p))$. SMC uses weighted particles and resampling to approximate such target distributions (Del Moral et al., 2006). In our search, intermediate rewards guide particle selection, and the terminal best-particle choice is a search decision rather than a draw from π.

At checkpoint m, let $w_m^{(k)}$ denote the nonnegative selection weight of arriving particle k. The effective sample size $\text{ESS}_m = (\sum_k w_m^{(k)})^2 / \sum_k (w_m^{(k)})^2$ measures weight concentration. We use the empirical squared coefficient of variation,

$$\hat{\sigma}_m^2 = \frac{K_{m-1}}{\text{ESS}_m} - 1, \tag{2}$$

to guide compute across checkpoints. The sum $\sum_m \hat{\sigma}_m^2 / K_{m-1}$ is a tractable variance *surrogate*, not the exact variance of the final output.

Metropolis-Hastings (MH). For a target distribution π, MH (Metropolis et al., 1953; Hastings, 1970) proposes $x' \sim q(\cdot \mid x)$ and accepts it with probability

$$\alpha(x, x') = \min\left\{1,\ \frac{\pi(x')\, q(x \mid x')}{\pi(x)\, q(x' \mid x)}\right\}. \tag{3}$$

Our proposal adds noise to the output latent and then re-denoises it along a new stochastic path. If the reverse process is exact, the acceptance rule for the reward-tilted target reduces to $\alpha = \min\{1, \exp(\lambda \Delta R)\}$ (Appendix A.6). With a learned reverse model, this reward-only acceptance rule is approximate. Under standard conditions, repeated exact MH steps converge to the target distribution, while a finite run also depends on its initialization and proposal scale.

4 METHOD

SWIM combines adaptive particle search with Metropolis-Hastings (MH) refinement under a fixed compute budget. Stage 1 uses an adaptive particle funnel (Section 4.1) to identify a promising candidate under budget B_{part}, while Stage 2 applies T MH proposals (Section 4.2) to further improve its reward. If proposal j costs c_j generator forward evaluations, the total budget is $B_{\text{total}} = B_{\text{part}} + \sum_{j=1}^{T} c_j$. The budget split between the two stages is fixed, while our online allocation strategy adapts computation within the particle stage.

4.1 ADAPTIVE PARTICLE FUNNEL

Formulation. Consider a diffusion or flow-matching model with N denoising steps and M fixed checkpoints at noise levels $t_1 > \cdots > t_M$. Let K_0 denote the initial particle population, and let K_m denote the population after resampling at checkpoint m. Let n_1 denote the number of steps before the first checkpoint. For $m < M$, let L_m denote the segment length between checkpoints m and $m{+}1$. Let L_M denote the final segment after checkpoint M. The cost to propagate K_m particles for L_m steps is $K_m L_m$ generator forward evaluations. At checkpoint m, we score the K_{m-1} arriving particles and compute the selection weights $w_k = \exp(\lambda\,(R_k - R_{\max}))$.

We measure the dispersion of these weights by their squared coefficient of variation. Its empirical estimate is $\widehat{\sigma}_m^2 = \frac{K_{m-1}}{\mathrm{ESS}_m} - 1$. This estimate is zero when all particles have equal weight. It increases as the weights become concentrated on fewer particles. We use this dispersion to define the estimation-variance surrogate $V_{\mathrm{sur}} = \sum_{m=1}^{M} \frac{\sigma_m^2}{K_{m-1}}$.

Here, σ_m^2 denotes the anticipated weight dispersion at checkpoint m, while $\widehat{\sigma}_m^2$ denotes its observed estimate. We treat the dispersion coefficients as fixed. We then choose the population sizes to minimize this surrogate under the particle budget.

We set $K_M = 1$ and define: $\tilde{B} = B_{\mathrm{part}} - L_M$. The continuous allocation problem is

$$\min_{K_1,\ldots,K_{M-1}} \sum_{m=1}^{M-1} \frac{\sigma_{m+1}^2}{K_m} \qquad \text{s.t.} \quad \sum_{m=1}^{M-1} K_m L_m = \tilde{B} - K_0 n_1. \tag{4}$$

We treat K_0 as a hyperparameter. It sets the initial exploration population before online dispersion estimates are available. We deduct its cost, $K_0 n_1$, from the budget. We allocate the remaining budget over $K_1, \ldots, K_{M-1}$. Let $B_{\mathrm{seg}} = \tilde{B} - K_0 n_1$ denote the budget available for $K_1, \ldots, K_{M-1}$.

Theorem 1 (Optimal allocation for the variance surrogate, proof in Appendix A.2)**.** *The solution to Equation* (4) *is*

$$K_m^* = \frac{\tilde{B} - K_0 n_1}{\mathcal{Z}} \cdot \frac{\sigma_{m+1}}{\sqrt{L_m}}, \qquad 1 \le m \le M{-}1, \tag{5}$$

where $\mathcal{Z} = \sum_{m=1}^{M-1} \sigma_{m+1}\sqrt{L_m}$.

The optimal allocation assigns more particles to stages with greater anticipated weight dispersion and fewer to longer, more expensive stages.

Online algorithm. Theorem 1 gives the optimal allocation when the future dispersions $\{\sigma_{m+1}\}$ are known. Its optimality condition can equivalently be written as: $K_m^* = \sigma_{m+1}/\sqrt{\nu L_m}$, where the multiplier ν sets the overall compute scale so that the allocations satisfy the budget.

At inference time, neither the future dispersion nor the corresponding multiplier is known. We use the current dispersion as a one-step forecast, $\hat{\sigma}_{m+1|m}^2 = \hat{\sigma}_m^2$. We then adapt the multiplier from the compute used so far. Before choosing K_m, let $C_{m-1} = \sum_{j=1}^{m-1} K_j L_j$ be the allocation compute already spent, and let $T_{m-1} = B_{\mathrm{seg}} \frac{\sum_{j=1}^{m-1} L_j}{\sum_{j=1}^{M-1} L_j}$ be the corresponding proportional-spending target. Starting from $\nu_1 = \nu_{\min}$, we set

$$\nu_m = \max\left\{\nu_{\min}, \nu_{m-1} + \eta_\nu \frac{C_{m-1} - T_{m-1}}{B_{\mathrm{seg}}}\right\}, \tag{6}$$

and choose $K_m \approx \sqrt{\hat{\sigma}_{m+1|m}^2 / \nu_m L_m}$. If previous segments used more compute than the proportional target, ν_m increases, and the next population shrinks. If they used less, ν_m decreases, and the next population can grow. When the population grows, the growth rule determines which particles are duplicated (Figure 12).

The step size η_ν controls how quickly the multiplier responds to budget error. Because the one-step dispersion forecast can be inaccurate, we clip the proposed population between a pacing floor K_{pace}, which preserves a fraction γ of the uniform allocation, and an affordability cap $K_m^{\max}$, which reserves enough compute for the remaining segments (Appendices A.3 and A.4).

Algorithm 1 SWIM: Particle Funnel and MH Refinement

Require: Prompt p, particle budget B_{part}, MH steps T, checkpoints $\{n_1, \dots, n_M\}$, rewards R_t, R_{MH}

Initialize K_0 particles from i.i.d. noise, set $\nu \leftarrow \nu_{\min}$, $B_{\text{rem}} \leftarrow B_{\text{part}}$
for denoising step $i = 0, \dots, N-1$ **do**
  if $i = n_m$ for some checkpoint m **then**
    Score all particles, $\hat{\sigma}_m^2 \leftarrow K_{\text{cur}}/\text{ESS} - 1$
    $\hat{\sigma}_{m+1|m}^2 \leftarrow \hat{\sigma}_m^2$
    Update ν, compute $K_m^{\max}$ and K_{pace}
    $K_m \leftarrow \text{clip}\Big(\text{round}\left(\sqrt{\hat{\sigma}_{m+1|m}^2/(\nu L_m)}\right), K_{\text{pace}}, K_m^{\max}\Big)$
    Resample to K_m particles
  end if
  Advance all particles by one denoising step
  $B_{\text{rem}} \leftarrow B_{\text{rem}} - K_{\text{cur}}$
end for
$x^* \leftarrow \arg\max_k R(x_0^{(k)})$
for MH step $j = 1, \dots, T$ **do**
  Propose $x' \leftarrow \text{NOISEDENOISE}(x^*, s_{\text{mh}})$
  Accept $x^* \leftarrow x'$ with probability $\min\{1, \exp(\lambda_{\text{MH}} \Delta R_{\text{MH}})\}$
end for
return x^*

4.2 METROPOLIS-HASTINGS REFINEMENT

The particle funnel returns the terminal particle with the highest particle-stage reward, which initializes a short MH chain. At each MH step, we re-noise its output latent to level s_{mh} and re-denoise both modalities along a fresh stochastic path to obtain a proposal x'. Thus, s_{mh} sets the proposal scale and cost, while T sets the number of refinement steps. We score each completed proposal with the fixed reward $R_{\text{MH}}(x) = \sum_{q=1}^{Q} \beta_q R_q(x)$, where R_q is an alignment or quality objective and β_q sets its weight. A proposal is accepted with probability $\alpha(x, x') = \min\{1, \exp(\lambda_{\text{MH}}[R_{\text{MH}}(x') - R_{\text{MH}}(x)])\}$. Under an exact reverse model, this reward-only rule is the MH acceptance ratio for the reward-tilted reference distribution. With a learned reverse model, it is approximate (Appendix A.6). Because we run only a short MH chain, its starting point matters. We therefore use particle search for global exploration and MH to refine its best terminal candidate (Appendix A.7).

Particle-stage reward. Particle search uses a checkpoint-dependent reward:

$$R_t(x) = R_1(x) + \beta_{\text{mix}} \mathbf{1}[t < t_{\text{mix}}] R_2(x).$$

Here, R_t is the reward used to score particles at denoising time t. The term R_1 is used at every checkpoint, while R_2 is added once $t < t_{\text{mix}}$, when its intermediate estimate is more reliable.

5 EXPERIMENTS

We report our experimental setup, including models and benchmarks, baselines, metrics, and implementation, our main quantitative and qualitative results, and our ablation studies.

5.1 SETUP

Models and benchmarks. We evaluate SWIM on two joint audio-visual models, JavisDiT++ (Liu et al., 2026b) and LTX-2 (HaCohen et al., 2026), on JAVISBENCH-MINI (Liu et al., 2026a). Beyond video generation, we also evaluate SWIM in the image domain with Stable Diffusion v1.5 (Rombach et al., 2022) on DrawBench (Saharia et al., 2022) using ImageReward (Xu et al., 2023) and on Simple Animals (Black et al., 2024) using AestheticScore (LAION-AI, 2022), following the image-domain evaluation setup of VASR (Shekhar et al., 2026).

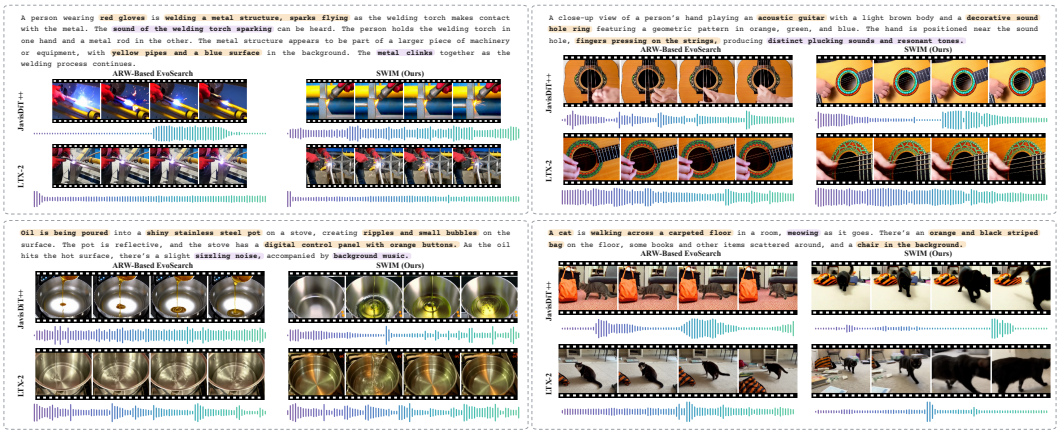


Figure 1: Visual results on JAVISBENCH-MINI (Liu et al., 2026a). For each prompt, we show the ARW-based EvoSearch and SWIM outputs for both JavisDiT++ and LTX-2, each as a frame strip with its audio waveform below. Cues in the prompts are highlighted as **visual** or **audio**.

Baselines. We compare SWIM with Best-of-N (BoN) and EvoSearch using the Adaptive Reward Weighting (ARW) scheme from ITS-AVGen (Jung et al., 2026), and with Uniform+MH. We also include Uniform with a fixed particle population and no MH refinement, and view BoN as MH initialized from full noise. BoN generates independent samples and returns the highest-scoring one. EvoSearch (He et al., 2025) performs evolutionary search along the denoising trajectory. Uniform+MH uses a uniform particle population at each checkpoint followed by MH refinement, isolating the effect of adaptive particle allocation. We report the NFE for each setting and compare methods at equal or closely matched budgets.

Metrics. For AV generation, we report text-visual ImageBind (TV-IB), text-audio ImageBind (TA-IB), audio-visual ImageBind (AV-IB) (Girdhar et al., 2023), AVHScore (AVH) (Mao et al., 2024), JavisScore (JS) (Liu et al., 2026a), CLAP text-audio (CLAP) (Wu et al., 2024), and VideoReward text-alignment (VR-TA) (Liu et al., 2025). For image generation, we report ImageReward (Xu et al., 2023) and AestheticScore (LAION-AI, 2022). Higher scores are better for all metrics.

Implementation. Unless otherwise specified, all experiments use $N{=}50$ denoising steps and checkpoints at steps $\{10, 20, 30, 40\}$. We set the particle-alignment temperature to $\lambda{=}1.0$. For joint audio-visual generation, the particle funnel uses the staged reward $R_t = R_{\text{tv}} + \beta_{\text{mix}}\mathbf{1}[t < t_{\text{mix}}]R_{\text{ta}}$, with $\beta_{\text{mix}}{=}0.3$ and $t_{\text{mix}}{=}0.85$. MH uses the fixed reward $R_{\text{MH}} = \text{JS} + \beta_{\text{MH}}\text{VR-TA}$ with $\beta_{\text{MH}}{=}0.05$, which improves alignment while discouraging loss of video quality. Motivated by the image-domain results in Table 2, we use systematic resampling for AV-generation. We set the pacing floor to $\gamma{=}0.9$. For joint audio-visual generation, we evaluate two total compute budgets: 360 NFE (200 NFE for particle search and 160 NFE for MH) and 560 NFE (400 NFE for particle search and 160 NFE for MH), with initial populations of $K_0{=}6$ and $K_0{=}8$, respectively. MH refinement uses $T{=}20$ proposals, each costing 8 NFE, with $\lambda_{\text{MH}}{=}150$ and $s_{\text{mh}}{=}0.7$.

5.2 MAIN RESULTS

Audio-visual generation. Table 1 compares methods at lower and higher compute budgets. We take the mean of relative gains across the seven metrics and use the stronger of Uniform+MH and SWIM for each comparison, which is SWIM in all four model-budget settings. At the lower, closely matched budget, SWIM improves over BoN and EvoSearch by 30.7% and 14.1% with JavisDiT++, and by 9.2% and 17.0% with LTX-2. Against Uniform, its mean relative gains are 7.0% with JavisDiT++ and 8.9% with LTX-2. At the higher budget, its gains over BoN and EvoSearch are 31.5% and 13.5% with JavisDiT++, and 7.2% and 10.9% with LTX-2. Comparing SWIM with Uniform+MH isolates the effect of adaptive allocation. The mean relative gains are 1.0% and 1.2% at the lower budget, and 1.2% and 3.0% at the higher budget for JavisDiT++ and LTX-2, respectively.

We also show visual examples on four sample prompts from JAVISBENCH-MINI (Liu et al., 2026a) in Figure 1, and include a larger set of 80 prompts in the supplementary material.

Image generation. Table 2 shows that SWIM improves over the corresponding fixed-population Uniform+MH baseline under systematic, multinomial, and Fleming-Viot resampling, indicating that its gains do not depend on the resampling strategy. Taking the mean across AestheticScore and ImageReward, the relative gains are 4.1% with systematic, 4.6% with multinomial, and 3.3% with Fleming-Viot resampling, for an overall mean of 4.0%.

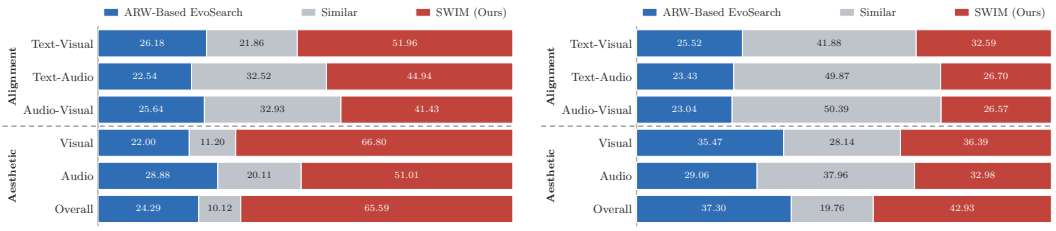
Table 1: Results on JAVISBENCH-MINI (Liu et al., 2026a). Budget is reported as total NFE (forward passes). SWIM achieves the strongest results at lower budgets, the practical regime for expensive AV models. Numbers reported as mean(std dev), ↑: higher is better.

Model	Method	TV-IB ↑	TA-IB ↑	AV-IB ↑	AVH ↑	JS ↑	CLAP ↑	VR-TA ↑
JavisDiT++	BoN@400	0.3780 (0.0002)	0.1496 (0.0006)	0.1859 (0.0015)	0.1711 (0.0016)	0.1021 (0.0015)	0.1474 (0.0015)	0.3711 (0.0015)
	BoN@600	0.3785 (0.0003)	0.1508 (0.0013)	0.1896 (0.0018)	0.1746 (0.0013)	0.1035 (0.0003)	0.1464 (0.0017)	0.3870 (0.0001)
	EvoSearch@399	0.3762 (0.0011)	0.1766 (0.0012)	0.2312 (0.0005)	0.2126 (0.0016)	0.1380 (0.0016)	0.1857 (0.0014)	0.3074 (0.0048)
	EvoSearch@532	0.3767 (0.0006)	0.1838 (0.0022)	0.2382 (0.0003)	0.2197 (0.0013)	0.1441 (0.0004)	0.1933 (0.0024)	0.3223 (0.0271)
	Uniform@400	**0.3824** (0.0004)	0.1861 (0.0026)	0.2218 (0.0016)	0.2098 (0.0019)	0.1125 (0.0023)	0.1868 (0.0066)	**0.6245** (0.0036)
	Uniform+MH@360	0.3755 (0.0010)	**0.1900** (0.0002)	0.2434 (0.0021)	0.2335 (0.0023)	0.1692 (0.0024)	0.1878 (0.0022)	0.4417 (0.0132)
	Uniform+MH@560	0.3786 (0.0002)	0.1885 (0.0042)	0.2408 (0.0032)	0.2304 (0.0033)	0.1649 (0.0004)	0.1894 (0.0008)	**0.5325** (0.0132)
	SWIM @360	0.3795 (0.0009)	0.1894 (0.0016)	**0.2442** (0.0018)	**0.2357** (0.0017)	**0.1695** (0.0013)	**0.1901** (0.0033)	0.4587 (0.0040)
	SWIM @560	**0.3789** (0.0004)	**0.1899** (0.0009)	**0.2448** (0.0020)	**0.2340** (0.019)	**0.1655** (0.0024)	**0.1973** (0.0012)	0.5299 (0.0136)
LTX-2	BoN@400	0.3797 (0.0008)	0.1598 (0.0021)	0.1984 (0.0014)	0.1823 (0.0007)	0.1055 (0.007)	0.1230 (0.0035)	0.4731 (0.0020)
	BoN@600	0.3802 (0.0003)	0.1625 (0.0011)	0.2061 (0.0020)	0.1905 (0.0014)	0.1042 (0.0019)	0.1394 (0.0012)	0.4983 (0.0034)
	EvoSearch@399	0.3775 (0.0001)	0.1633 (0.0002)	0.2044 (0.0015)	0.1394 (0.0009)	0.1014 (0.0039)	0.1421 (0.0016)	0.3571 (0.0028)
	EvoSearch@532	0.3778 (0.0006)	0.1688 (0.0011)	0.2109 (0.0014)	0.1949 (0.0013)	0.1095 (0.0022)	0.1484 (0.0036)	0.3643 (0.0006)
	Uniform@400	**0.3823** (0.0001)	0.1691 (0.0024)	0.2027 (0.0006)	0.1874 (0.0005)	0.0686 (0.0018)	0.1378 (0.0034)	**0.6729** (0.0031)
	Uniform+MH@360	0.3796 (0.0002)	0.1698 (0.0013)	0.2106 (0.0046)	0.1995 (0.0042)	0.1081 (0.0025)	0.1453 (0.0048)	0.5329 (0.0071)
	Uniform+MH@560	0.3815 (0.0001)	0.1672 (0.0028)	0.2099 (0.0033)	0.1974 (0.0028)	0.1058 (0.0027)	0.1328 (0.0029)	0.6165 (0.0096)
	SWIM @360	0.3799 (0.0002)	**0.1701** (0.006)	**0.2111** (0.0007)	**0.2028** (0.0002)	**0.1105** (0.0016)	**0.1497** (0.0047)	0.5387 (0.0106)
	SWIM @560	**0.3822** (0.0006)	**0.1696** (0.0023)	**0.2114** (0.0060)	**0.2004** (0.0049)	**0.1102** (0.0017)	**0.1498** (0.0002)	**0.6193** (0.0003)

Table 2: Results on DrawBench (ImageReward) (Saharia et al., 2022) and Simple Animals (AestheticScore) (Black et al., 2024). SWIM outperforms fixed-population baselines across all resampling strategies at matched NFE. Numbers reported as mean(std dev), ↑: higher is better.

Method	SimpleAnimals: AestheticScore ↑	DrawBench: ImageReward ↑
Uniform+MH(Sys)@400	6.506 (0.018)	0.9890 (0.027)
Uniform+MH(Mul)@400	6.395 (0.016)	0.9129 (0.030)
Uniform+MH(FV)@400	6.452 (0.024)	0.9677 (0.023)
SWIM-Sys@400	**6.652** (0.021)	**1.048** (0.022)
SWIM-Mul@400	**6.528** (0.029)	**0.978** (0.031)
SWIM-FV@400	**6.602** (0.017)	**1.010** (0.025)

5.3 HUMAN EVALUATION



(a) JavisDiT++ (b) LTX-2

Figure 2: Human preference responses comparing SWIM JavisDiT++ and LTX-2 on JAVISBENCH-MINI (Liu et al., 2026a). Bars show relative preferences on text-visual, text-audio, and audio-visual alignments, as well as visual, audio, and overall aesthetics. SWIM is preferred over the corresponding baseline across all the preference criteria.

The reported human evaluation compares adaptive SWIM with ARW-based EvoSearch, with each video generated on a budget of 560 NFE. The study comprises 80 prompts from JAVISBENCH-MINI (Liu et al., 2026a) (all available in the supplementary). We recruited 62 participants in total and showed each participant 25 random prompts from these 80 along with two videos, one using ARW-based EvoSearch and the other using SWIM, in random order. In each pair, the base model was

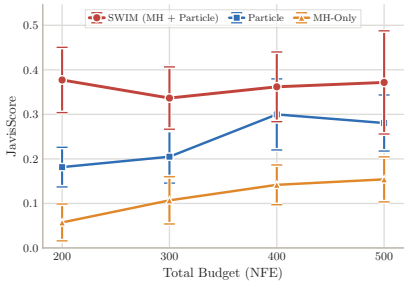


Figure 3: Particle-only, MH-only, and SWIM at matched total NFE. SWIM has the highest mean JavisScore at each tested budget, with its largest mean advantage over particle-only search at the lowest budget.

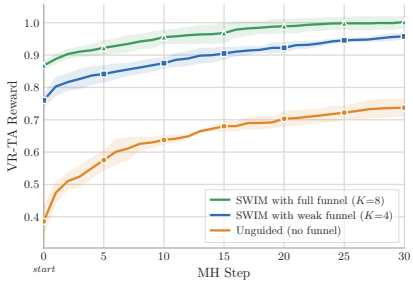
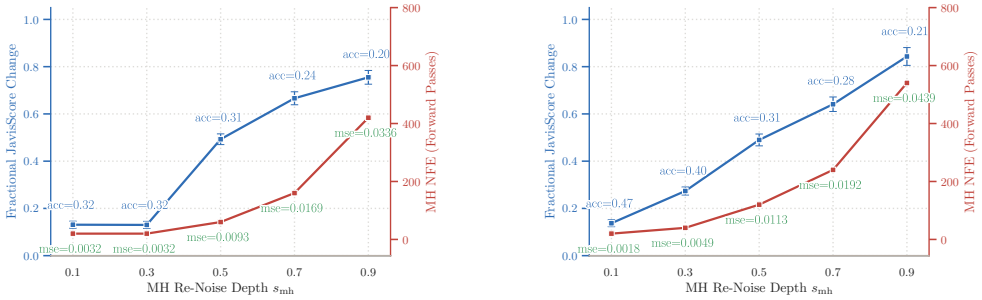


Figure 4: VR-TA reward over $T{=}30$ MH steps from an unguided sample, a weak funnel ($K{=}4$), and a full funnel ($K{=}8$), with $s_{\text{mh}}{=}0.7$. Stronger starts reach higher final reward but leave less improvement for MH.

either JavisDiT++ or LTX-2, chosen randomly and independently across pairs. Raters judged each pair on three alignment criteria, text-visual, text-audio, and audio-visual, and three aesthetic criteria, visual, audio, and overall. Figure 2 reports the resulting preference shares, pooled over 741 pairs of JavisDiT++ videos and over 764 pairs of LTX-2 videos. SWIM outperforms both JavisDiT++ and LTX-2 baselines across all six preference criteria, with preference margins of $15 - 44\%$ over JavisDiT++ and $1 - 7\%$ over LTX-2.

5.4 Ablations



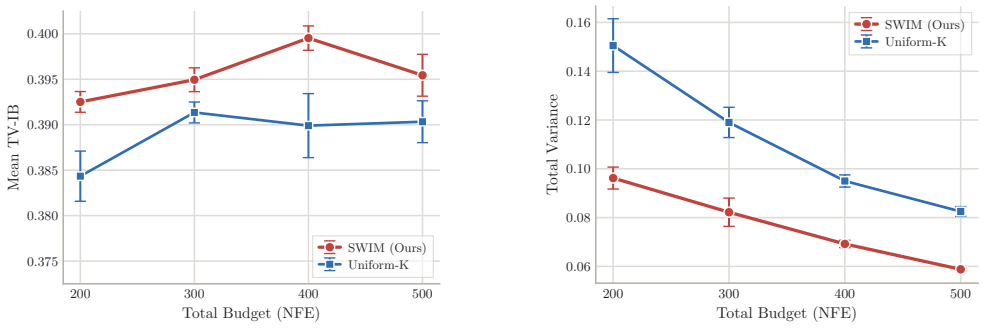
(a) JavisDiT++ (b) LTX-2

Figure 5: Effect of MH noise level s_{mh} on JavisDiT++ (left) and LTX-2 (right). Bars show relative JavisScore improvement over the funnel-only baseline and lines show MH NFE cost. The reported setting uses $s_{\text{mh}} = 0.7$.

Particle search versus MH refinement. Figure 3 compares relative improvements in JavisScore for particle-only search, MH-only refinement, and SWIM at matched NFE. Results on JavisDiT++ are means over 50 prompts and 3 seeds. MH-only has the smallest relative improvement, as starting from an unguided candidate requires substantial compute for effective mixing. Particle-only search achieves larger improvements through global exploration but remains limited by finite-population estimation error, especially at low budgets. SWIM achieves the highest relative improvement by using particle search to find a promising candidate and MH to refine it. Its advantage over particle-only search is largest at low budgets and narrows as compute increases, consistent with MH correcting some estimation error left by a smaller particle population.

Quality of the MH initialization. Figure 4 compares MH-only, a weak funnel ($K = 4$) followed by MH, and a stronger funnel ($K = 8$) followed by MH. Results on JavisDiT++ are means over 50 prompts and 3 seeds. All settings use the same VR-TA reward and MH schedule. MH-only improves initially but quickly plateaus below both funnel-initialized chains. The stronger funnel

starts highest and peaks earlier. The weaker funnel starts lower but gains more during MH, closing much of the gap and reaching a comparable final reward with fewer particles. This suggests that a strong start accelerates reward gains, while MH can recover some of the advantage of a larger particle population.



(a) TV-IB vs. NFE

(b) Variance vs. NFE

Figure 6: TV-IB and estimator-variance surrogate versus NFE on JavisDiT++. The left panel shows TV-IB and the right panel shows the variance surrogate across compute budgets. Adaptive allocation achieves higher TV-IB and lower variance than uniform-K throughout the tested range.

Effect of s_{mh} in MH refinement. In the MH phase, s_{mh} sets the noise level used to perturb the latent before re-denoising, and thus controls proposal size. Figure 5 sweeps $s_{\text{mh}} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ on JavisDiT++ and LTX-2, reporting JavisScore improvement over the funnel-only baseline and MH NFE cost. Results are means over 50 prompts and 3 seeds per model. At low s_{mh} (≤ 0.3), proposals are conservative and yield small improvements despite high acceptance. At high values (≥ 0.8), re-denoising from near-pure noise can discard the funnel's coarse alignment, making proposals resemble independent generations rather than local refinements. Around $s_{\text{mh}} = 0.7$, proposals improve audio-visual timing while preserving structure, balancing useful changes against destructive proposals. We use this setting in subsequent experiments.

Performance and variance across budgets. We use VR-TA to guide particle allocation for JavisDiT++ and report TV-IB as a separate evaluation metric to avoid tuning and evaluating on the same reward (Figure 6). Results are means over 50 prompts and 3 seeds. Adaptive allocation yields higher TV-IB and lower estimator variance than uniform-K across budgets. Score gains are smallest at the extremes: little compute can be redistributed at low budgets, while uniform allocation closes part of the gap at high budgets. The variance gap also narrows at the largest budget but remains visible. SD1.5 shows a similar pattern, with smaller quality and variance benefits as NFE increases (Figure 11).

6 Limitations and Conclusion

We present SWIM as a two-stage method for efficient inference-time alignment. Stage 1 uses variance-optimal particle allocation to search for a candidate. Stage 2 uses Metropolis-Hastings (MH) to refine it locally, addressing reward-estimation error from finite particles. Online ESS guides allocation toward informative denoising stages. Together, particle search and MH combine global exploration with local refinement within a limited budget, without offline calibration or reward-specific tuning. The allocation theorem $K_m^* \propto \sigma_m / \sqrt{L_m}$ is model- and reward-agnostic and applies to diffusion and flow-matching models. At larger budgets, the smaller gains from adaptive allocation may not justify its extra schedule parameters. That being said, large models still incur substantial inference cost. Step distillation, latent caching, or token merging could reduce per-step cost alongside our budget-aware scheduling. The persistence forecast $\hat{\sigma}^2_{m+1|m} = \hat{\sigma}^2_m$ may be suboptimal when variance changes rapidly between adjacent checkpoints.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2024. URL https://arxiv.org/abs/2305.13301.

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, 2023. URL https://arxiv.org/abs/2311.15127.

Xin Cheng, Xihua Wang, Ying Ba, Yuyue Wang, Kaisi Guan, Yinbo Wang, Wenpu Li, and Ruihua Song. Syncdpo: Enhancing temporal synchronization in video-audio joint generation via preference learning, 2026. URL https://arxiv.org/abs/2605.12179.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards, 2024. URL https://arxiv.org/abs/2309.17400.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006. doi: 10.1111/j.1467-9868.2006.00553.x. URL https://academic.oup.com/jrsssb/article-abstract/68/3/411/7110641.

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

Zach Evans, CJ Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion, 2024. URL https://arxiv.org/abs/2402.04825.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all, 2023. URL https://arxiv.org/abs/2305.05665.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Guetta, Noa Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, Victor Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. Ltx-2: Efficient joint audio-visual foundation model, 2026. URL https://arxiv.org/abs/2601.03233.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444, 14643510. doi: 10.1093/biomet/57.1.97. URL http://www.jstor.org/stable/2334940.

Haoran He, Jiajun Liang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Ling Pan. Scaling image and video generation via test-time evolutionary search, 2025. URL https://arxiv.org/abs/2505.17618.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models, 2025. URL https://arxiv.org/abs/2506.20701.

Jaemin Jung, Kyeongha Rho, Inkyu Shin, and Joon Son Chung. Inference-time scaling for joint audio-video generation, 2026. URL https://arxiv.org/abs/2606.03183.

LAION-AI. Aesthetic predictor, 2022. URL https://github.com/LAION-AI/aesthetic-predictor.

Sanghyun Lee, Sunwoo Kim, Seungryong Kim, Jongho Park, and Dongmin Park. Effective test-time scaling of discrete diffusion through iterative refinement, 2025. URL https://arxiv.org/abs/2511.05562.

Sen Liang, Cong Wang, Fengbin Guan, Zhentao Yu, Yiting Lu, Yuanzhi Wang, Yuan Zhou, Xin Li, and Zhibo Chen. Spongebob: Sync-aware harmonious audio-visual generative editing, 2026. URL https://arxiv.org/abs/2605.25193.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models, 2023. URL https://arxiv.org/abs/2301.12503.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di ZHANG, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 82155–82192. Curran Associates, Inc., 2025. doi: 10.52202/085713-2750. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/76227feb18ea0ee40bd15cf02c33e18e-Paper-Conference.pdf.

Kai Liu, Wei Li, Lai Chen, Shengqiong Wu, Yanhao Zheng, Jiayi Ji, Fan Zhou, Jiebo Luo, Ziwei Liu, Hao (Scofield) Fei, and Tat-Seng Chua. Javisdit: Joint audio-video diffusion transformer with hierarchical spatio-temporal prior synchronization. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 139160–139194, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e11afe29671baa67c9f15fa77fc97357-Paper-Conference.pdf.

Kai Liu, Yanhao Zheng, Kai Wang, Shengqiong Wu, Rongjunchen Zhang, Jiebo Luo, Dimitrios Hatzinakos, Ziwei Liu, Hao Fei, and Tat-Seng Chua. Javisdit++: Unified modeling and optimization for joint audio-video generation. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/pdf?id=hRRWfFpKRp.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Yuxin Mao, Xuyang Shen, Jing Zhang, Zhen Qin, Jinxing Zhou, Mochu Xiang, Yiran Zhong, and Yuchao Dai. Tavgbench: Benchmarking text to audible-video generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pp. 6607–6616, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706868. doi: 10.1145/3664647.3680612. URL https://doi.org/10.1145/3664647.3680612.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997. ISSN 10505164. URL http://www.jstor.org/stable/2245134.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models, 2024. URL https://arxiv.org/abs/2406.07524.

Shivanshu Shekhar and Tong Zhang. Rocm: Rlhf on consistency models, 2025. URL https://arxiv.org/abs/2503.06171.

Shivanshu Shekhar, Shreyas Singh, and Tong Zhang. See-dpo: Self entropy enhanced direct preference optimization, 2025. URL https://arxiv.org/abs/2411.04712.

Shivanshu Shekhar, Sagnik Mukherjee, Jia Yi Zhang, and Tong Zhang. Vasr: Variance-aware systematic resampling for reward-guided diffusion, 2026. URL https://arxiv.org/abs/2604.06779.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models, 2025. URL https://arxiv.org/abs/2501.06848.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts, 2025. URL https://arxiv.org/abs/2503.02819.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

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, 2025. URL https://arxiv.org/abs/2503.20314.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation, 2024. URL https://arxiv.org/abs/2211.06687.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023. URL https://arxiv.org/abs/2304.05977.

AI USE STATEMENT

Generative AI tools assisted with manuscript framing and editing, identifying and checking literature references, reviewing mathematical claims and proofs, and providing feedback on methodology and interpretation of reported results. The authors are responsible for the final content, including AI-assisted text and claims.

A Proofs and Derivations

A.1 Why the ESS-Based Objective Is Reasonable

The squared coefficient of variation of checkpoint weights is estimated by $\hat{\sigma}_m^2 = K_{m-1}/\mathrm{ESS}_m - 1$. If K weights were independent draws from a fixed incremental-weight distribution, the relative variance of their sample mean would equal its population squared coefficient of variation divided by K. This elementary calculation motivates the σ_m^2/K_{m-1} terms in the allocation surrogate. Resampled trajectories share ancestors, however, so the full SMC variance generally includes dependence and propagation effects (Del Moral et al., 2006; Shekhar et al., 2026). We do not identify the surrogate with the exact variance of the final output.

Weight dispersion need not indicate a useful reward signal. A noisy proxy can separate particles strongly while predicting final quality poorly. For a fixed population and a small reward temperature, exponentially reweighting by an intermediate reward R_m changes the mean of a terminal reward R_0 to first order by $\lambda \operatorname{Cov}(R_0, R_m)$. This is a local approximation, not an exact finite-population resampling formula. It explains why the staged reward delays the text-audio term until its intermediate estimate becomes more useful. The online allocator uses weight dispersion because it is available at each checkpoint. The experiments assess whether that proxy helps final quality.

A.2 Optimal Allocation

Proof of Theorem 1. With K_0 fixed, let $B_{\mathrm{seg}} = \tilde{B} - K_0 n_1$ be the budget available to the decision variables. Introduce the Lagrange multiplier $\nu \in \mathbb{R}$ for the compute constraint. The Lagrangian is

$$\mathcal{L} = \sum_{m=1}^{M-1} \frac{\sigma_{m+1}^2}{K_m} + \nu \left(\sum_{m=1}^{M-1} K_m L_m - B_{\mathrm{seg}} \right). \tag{7}$$

Stationarity with respect to each K_m gives

$$-\frac{\sigma_{m+1}^2}{K_m^2} + \nu L_m = 0 \quad \Longrightarrow \quad K_m^* = \frac{\sigma_{m+1}}{\sqrt{\nu L_m}}. \tag{8}$$

Substituting into the budget constraint gives

$$\frac{1}{\sqrt{\nu}} \sum_{m=1}^{M-1} \sigma_{m+1} \sqrt{L_m} = B_{\mathrm{seg}}. \tag{9}$$

Thus $\sqrt{\nu} = \mathcal{Z}/B_{\mathrm{seg}}$, where $\mathcal{Z} = \sum_{m=1}^{M-1} \sigma_{m+1}\sqrt{L_m}$. Substitution gives

$$K_m^* = \frac{B_{\mathrm{seg}}}{\mathcal{Z}} \frac{\sigma_{m+1}}{\sqrt{L_m}}. \tag{10}$$

The objective is convex for $K_m > 0$, so these stationary values minimize it under the budget constraint. □

Remark 1 (Continuous limit)**.** *As $M \to \infty$ and $L_m \to dt$, the problem becomes $\min_{K(\cdot)} \int_0^1 \sigma^2(t)/K(t)\, dt$ subject to $\int_0^1 K(t)\, dt = B$. The same pointwise Lagrangian argument gives $K^*(t) = B\, \sigma(t)/\|\sigma\|_{L^1}$.*

A.3 Uniform Pacing as a Minimax Safeguard

Consider J remaining segments with lengths L_j and allocations K_j, subject to $\sum_{j=1}^J L_j K_j \le B_{\mathrm{rem}}$. Suppose their unknown variance coefficients satisfy $0 \le \sigma_j^2 \le \bar{\sigma}^2$, and measure worst-stage risk by $W(K) = \max_j \sigma_j^2/K_j$. The uniform population allocation minimizes the worst-case risk.

Proof. For any allocation, the worst case is $\sup_{\sigma_j^2 \le \bar{\sigma}^2} W(K) = \bar{\sigma}^2 / \min_j K_j$. Also,

$$\min_j K_j \le \frac{\sum_j L_j K_j}{\sum_j L_j} \le \frac{B_{\mathrm{rem}}}{\sum_j L_j}. \tag{11}$$

Thus the worst-case risk is at least $\bar{\sigma}^2 \sum_j L_j / B_{\mathrm{rem}}$, attained by setting $K_j = B_{\mathrm{rem}} / \sum_j L_j$ for every segment. This is the continuous relaxation, and integer populations are handled by rounding and the affordability cap. □

A.4 Budget Feasibility

Theorem 2. *Assume $B \geq \sum_{j=1}^{M} L_j$ (the budget suffices for at least one particle per segment). If at every checkpoint $K_m \leq K_m^{\max} = \lfloor (B_m - \sum_{j>m} L_j)/L_m \rfloor$, then $B_{m+1} \geq \sum_{j>m} L_j$ for all m.*

Proof. By induction. Base: $B_1 \geq \sum_{j=1}^{M} L_j$ by assumption. Step: $K_m \leq K_m^{\max}$ gives

$$L_m K_m \leq B_m - \sum_{j>m} L_j, \tag{12}$$

so $B_{m+1} = B_m - L_m K_m \geq \sum_{j>m} L_j$. □

A.5 Reward Reliability Under Gaussian Assumptions

The following example shows how reward estimates can become more informative as denoising progresses. It does not determine the variance of particle-selection weights.

Proposition 1. *Let $x_0 \sim \mathcal{N}(0, \sigma_0^2 I_d)$, $x_t = (1-t)x_0 + t\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_d)$, and $R(x) = a^\top x + c$, where $a \neq 0$. Define reward reliability as $\rho(t) = \mathrm{Corr}(R(\mathbb{E}[x_0 \mid x_t]), R(x_0))$. Then*

$$\rho(t) = \frac{(1-t)\sigma_0}{\sqrt{(1-t)^2\sigma_0^2 + t^2}}. \tag{13}$$

Proof. The posterior $x_0 \mid x_t$ is Gaussian with covariance $\Sigma_{0|t} = t^2\sigma_0^2/((1-t)^2\sigma_0^2 + t^2) \cdot I_d$. For the linear reward, $\mathrm{Var}[R(\hat{x}_0) - R(x_0)] = \|a\|^2 t^2 \sigma_0^2/((1-t)^2\sigma_0^2 + t^2)$ and $\mathrm{Var}[R(x_0)] = \sigma_0^2\|a\|^2$. Then $\rho^2 = 1 - t^2/((1-t)^2\sigma_0^2 + t^2)$. □

Thus the denoised reward becomes more correlated with the terminal reward as t decreases in this Gaussian example. Checkpoint weight dispersion also depends on the reward, selection temperature, propagation, and shared particle ancestry, so this calculation does not predict a universal variance profile. The online scheduler uses the observed ESS at each checkpoint instead of assuming one.

A.6 MH Acceptance for Noising-Denoising Proposals

We derive the simplified acceptance ratio used in the MH refinement phase.

Setup. The ideal target is $\pi(x) \propto p_{\mathrm{ref}}(x) \exp(\lambda R(x))$. The proposal $q(x' \mid x)$ noises the entire latent to level s_{mh} and re-denoises:

1. Sample $x_s \sim p(x_s \mid x)$ by running the forward process to noise level s_{mh}.
2. Sample $x' \sim p_{\mathrm{rev}}(x' \mid x_s)$ by running the reverse SDE from s_{mh} to 0 with fresh Brownian motion.

The marginal proposal density is $q(x' \mid x) = \int p(x_s \mid x)\, p_{\mathrm{rev}}(x' \mid x_s)\, dx_s$.

Detailed balance of the bridge. If the reverse dynamics give the exact Bayes posterior of the forward noising process under p_{ref}, then $p_{\mathrm{rev}}(\cdot \mid x_s) = p_{\mathrm{ref}}(\cdot \mid x_s)$. The joint density over the triplet (x, x_s, x') is

$$p(x, x_s, x') = p_{\mathrm{ref}}(x)\, p(x_s \mid x)\, p_{\mathrm{ref}}(x' \mid x_s). \tag{14}$$

By Bayes' rule, $p_{\mathrm{ref}}(x)\, p(x_s \mid x) = p_s(x_s)\, p_{\mathrm{ref}}(x \mid x_s)$. Substituting:

$$p(x, x_s, x') = p_s(x_s)\, p_{\mathrm{ref}}(x \mid x_s)\, p_{\mathrm{ref}}(x' \mid x_s). \tag{15}$$

This is symmetric in x and x': both are conditionally independent given x_s, drawn from the same posterior. Marginalizing over x_s:

$$p_{\mathrm{ref}}(x)\, q(x' \mid x) = p_{\mathrm{ref}}(x')\, q(x \mid x'). \tag{16}$$

Cancellation. The MH acceptance is

$$\begin{aligned}\alpha &= \min\left\{1,\ \frac{\pi(x')\,q(x \mid x')}{\pi(x)\,q(x' \mid x)}\right\} \\ &= \min\left\{1,\ \frac{p_{\mathrm{ref}}(x')e^{\lambda R(x')}}{p_{\mathrm{ref}}(x)e^{\lambda R(x)}} \cdot \frac{p_{\mathrm{ref}}(x)}{p_{\mathrm{ref}}(x')}\right\} \\ &= \min\left\{1,\ e^{\lambda(R(x')-R(x))}\right\},\end{aligned} \tag{17}$$

where the second equality uses Equation (16). The p_{ref} factors cancel exactly.

Approximate case. With a learned model, $p_{\mathrm{rev}}(\cdot \mid x_s)$ only approximates $p_{\mathrm{ref}}(\cdot \mid x_s)$, so Equation (16) need not hold exactly. The simplified reward-only acceptance ratio is therefore approximate, and the resulting chain is not guaranteed to have the exact reward-tilted reference distribution as its stationary distribution.

A.7 Ideal Common-Target MH Refinement

Let π be a common target for both stages and P an exact MH kernel that leaves π invariant. Let μ_B be the unconditional distribution of the MH starting state after a particle budget B. This distribution includes the randomness of generation, resampling, and terminal selection. For any $T \geq 0$, Markov kernels contract total variation, so

$$\|\mu_B P^T - \pi\|_{\mathrm{TV}} = \|(\mu_B - \pi)P^T\|_{\mathrm{TV}} \leq \|\mu_B - \pi\|_{\mathrm{TV}}. \tag{18}$$

If P additionally contracts zero-mass signed measures by a factor $\rho < 1$, the right-hand side is at most $\rho^T\|\mu_B - \pi\|_{\mathrm{TV}}$. An invariant, irreducible, and aperiodic kernel converges under its usual regularity conditions, but its rate need not be useful at a short inference budget. Equation (18) explains why a closer initialization can help under a shared target, but it does not prove that selecting the highest-reward particle makes μ_B closer to π.

For a proposal q_s with acceptance probability α_s, stationary acceptance is $A_\pi(s) = \mathbb{E}_{X\sim\pi, Y\sim q_s(\cdot|X)}[\alpha_s(X, Y)]$. It depends on the target and proposal, not the chain's initialization. A larger proposal can travel farther when accepted yet also have lower acceptance. Therefore, neither acceptance alone nor the jump radius alone determines finite-budget mixing. In our practical method, the particle and MH rewards differ and the learned reverse proposal may not preserve π. The bound above is therefore a design motivation, while the reported output scores characterize the implemented procedure.

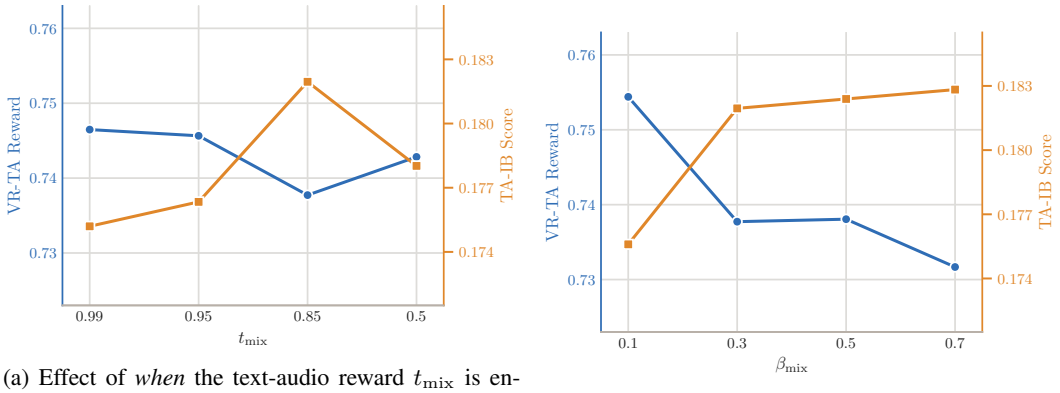
B Additional Experiments

B.1 Staged Reward Ablations

The staged reward in Section 4.2 introduces text-audio alignment at a selected denoising threshold t_{mix}, with its strength controlled by β_{mix}. Figure 7 examines the effect of both parameters. Introducing the text-audio reward too early ($t_{\mathrm{mix}} = 0.95$ or 0.99) leads to unreliable reward estimates at high noise levels and poor particle selection, consistent with the weak early-checkpoint discrimination of TA-IB in Figure 13, and degrades final alignment. Conversely, introducing it too late ($t_{\mathrm{mix}} = 0.50$) leaves insufficient search time for the text-audio reward to influence the final output. We find $t_{\mathrm{mix}} = 0.85$ provides the best balance between reliable reward estimates and sufficient search time. For reward strength, a small weight ($\beta_{\mathrm{mix}} = 0.1$) underweights text-audio alignment, resulting in lower TA-IB despite higher VR-TA. Increasing it to $\beta_{\mathrm{mix}} = 0.3$ substantially improves TA-IB with only a modest reduction in VR-TA, while larger weights (0.5 or 0.7) offer diminishing TA-IB gains at the expense of VR-TA. Although performance differences across these settings are relatively modest, the trends favor $t_{\mathrm{mix}} = 0.85$ and $\beta_{\mathrm{mix}} = 0.3$, which we use in all subsequent experiments.

B.2 Pacing Floor γ

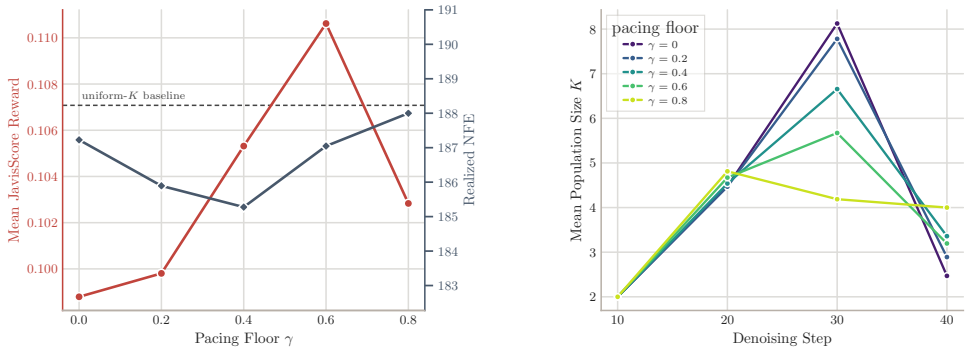
The online allocator uses a persistence forecast for the next checkpoint's variance. This forecast is useful when neighboring checkpoints have similar reward informativeness, but it can overreact



(a) Effect of *when* the text-audio reward t_{mix} is enabled.

(b) Effect of β_{mix} *after* t_{mix} is enabled.

Figure 7: Effect of the staged reward parameters. The left panel shows the effect of when the text-audio reward is enabled, and the right panel shows its weight β_{mix}. An intermediate activation threshold and moderate weight give the best balance between TA-IB and VR-TA.



(a) Final reward and realized NFE.

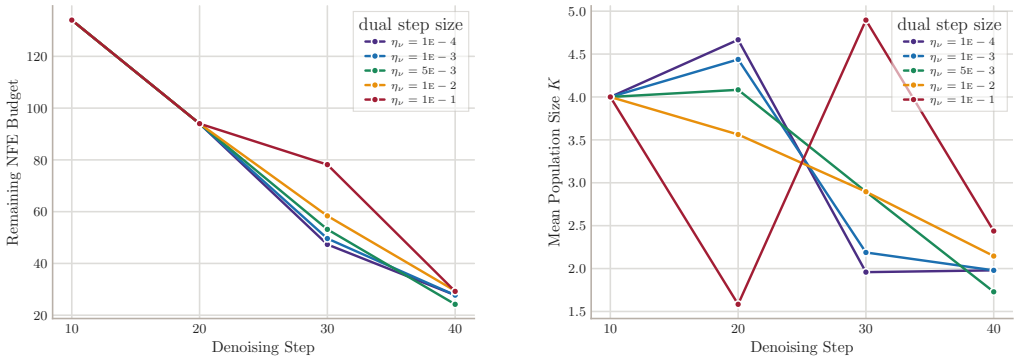
(b) Particle schedule across checkpoints.

Figure 8: Effect of the pacing floor γ on the adaptive funnel. Left: final reward and realized NFE. Right: actual particle schedule across checkpoints. Larger γ forces the schedule toward a more uniform allocation, and intermediate pacing achieves the best reward.

to a single high-variance stage and spend too much budget before later checkpoints. The pacing floor $K_{\text{pace}} = \lceil \gamma B_{\text{rem}} / \sum_{j \geq m} L_j \rceil$ prevents this failure mode by reserving a minimum particle population for future stages. Figure 8 shows the expected tradeoff. With small γ, the schedule is highly adaptive and places many particles at the informative middle checkpoint, but it can prune too aggressively afterward. With large γ, the schedule becomes closer to uniform and loses some of the benefit of adaptation. The best reward occurs at $\gamma = 0.6$, indicating that the optimal behavior lies between a fully adaptive persistence forecast and a uniform allocation.

B.3 Effect of Dual Step Size η_ν

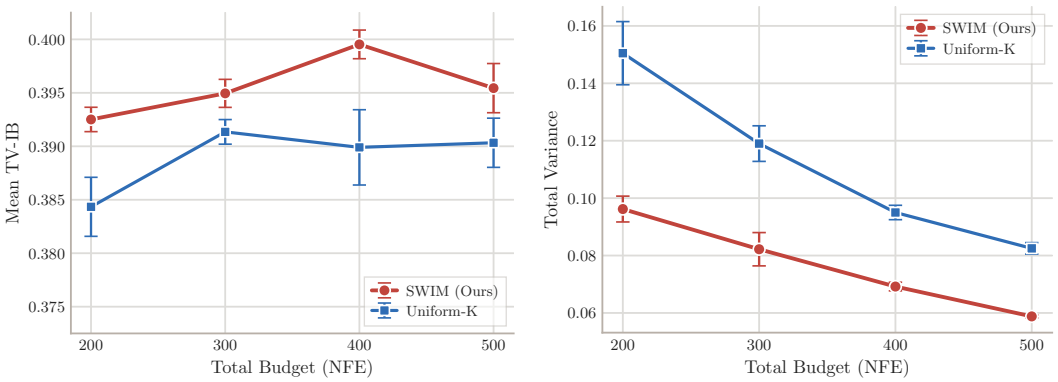
The dual variable ν acts as the shadow price of compute: a larger ν induces smaller populations, while η_ν controls how quickly it adapts to budget over- or under-spending. Setting $\eta_\nu = 0$ freezes ν at initialization, whereas larger values enable faster adaptation. Figure 9 sweeps $\eta_\nu \in \{1\text{E}-4, 1\text{E}-3, 5\text{E}-3, 1\text{E}-2, 1\text{E}-1\}$ on JavisDiT, with all other parameters fixed. Because ν is initialized from the near-zero variance at the first checkpoint and the dual update only acts thereafter, the schedule is identical across η_ν through the first two checkpoints and diverges only later. From there, larger η_ν prunes more aggressively, as it is more responsive to the previous stage and rapidly increases or decreases K, while small η_ν keeps K near K_0 and decreases it gradually. Total NFE stays near the cap for all values. Final performance follows a non-monotonic trend, with the highest reward at



(a) Remaining NFE budget.

(b) Mean population size K.

Figure 9: Effect of the dual step size η_ν on the funnel's compute schedule under JavisDiT++. The left panel shows remaining NFE budget and the right panel shows mean population size K across checkpoints. Larger η_ν responds more aggressively to budget errors, while schedules remain identical until the first update to ν.



(a) TV-IB vs. NFE.

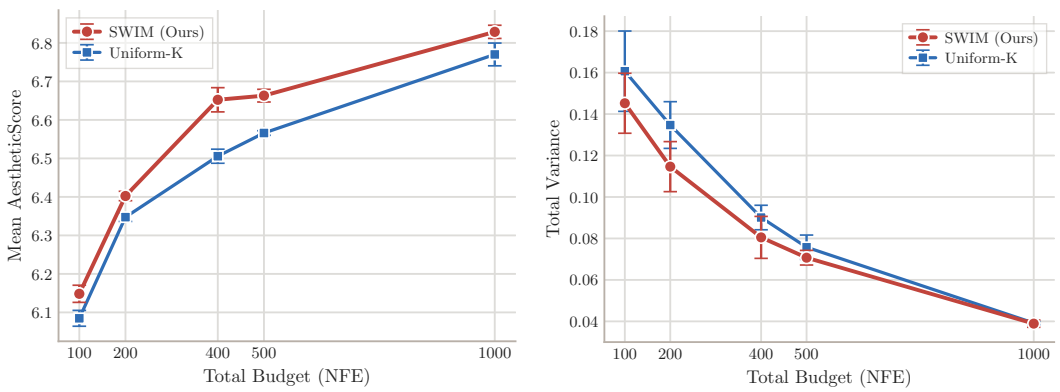
(b) Estimator variance vs. NFE.

Figure 10: Score and estimator variance under compute scaling for TV-IB in audio-visually aligned video generation. At full budget (NFE≈500), both methods perform comparably on score but SWIM retains a substantial advantage on variance. As budget decreases, SWIM maintains higher quality with estimator variance increasing comparatively smoothly, while uniform-K degrades more sharply. This behavior results from concentrating pruning at low-value checkpoints while preserving particles at informative middle checkpoints. Results on JavisDiT++ with 50 prompts and 3 seeds. Error bars show ± 1 standard error across prompts.

$\eta_\nu = 10^{-3}$ (0.215) and the lowest at $\eta_\nu = 10^{-1}$ (0.145). Moderate adaptation preserves particles until later denoising stages, where reward estimates are more informative, while overly aggressive adaptation causes frequent pruning and population growth, degrading performance.

B.4 Variance and Performance Curves for Stable Diffusion

Effect of compute scaling. Against corresponding uniform-K baselines, Figure 10a compares the quality-compute tradeoff of SWIM on JavisDiT++ using TV-IB across NFE budgets from 200 to 500, and Figure 11a compares the quality-compute tradeoff of SWIM on SD1.5 using AestheticScore across NFE budgets from 100 to 1000. SWIM consistently outperforms uniform-K across all budgets. On TV-IB, the largest improvement of 2.5% is observed at NFE=400, and on AestheticScore, the largest improvement of 2.3% is also observed at NFE=400. As compute increases, the score gap narrows to 1.3% at NFE=500 for TV-IB and to 0.9% at NFE=1000 for AestheticScore. This trend is expected, as larger particle populations allow even uniform allocation to explore the



(a) AestheticScore vs. NFE.

(b) Estimator variance vs. NFE.

Figure 11: Score and estimator variance under compute scaling for AestheticScore in image generation. At full budget (NFE≈1000), both methods perform comparably. As budget decreases, SWIM maintains higher quality with estimator variance increasing comparatively smoothly, while uniform-K degrades more sharply. This behavior results from concentrating pruning at low-value checkpoints while preserving particles at informative middle checkpoints. Results on SD1.5 with 45 prompts and 3 seeds. Error bars show ± 1 standard error across prompts.

reward landscape more effectively, reducing the benefit of adaptive allocation. The gains at low budgets are particularly relevant for expensive AV models, where the number of model evaluations is a major constraint.

Variance comparison. Figure 10b and Figure 11b compare the respective total estimator variances $\sum_m \hat{\sigma}_m^2/K_m$ for TV-IB and AestheticScore under the same compute budgets. SWIM achieves lower variance across all budgets. At budgets of NFE=200 and NFE=400, the respective reductions are 36.1% and 27.2% for TV-IB and 14.9% and 10.6% for AestheticScore. As compute increases, the variance gap remains at 28.7% at NFE=500 for TV-IB, while it narrows to 0.8% at NFE=1000 for AestheticScore. The trend for AestheticScore mirrors the score improvements in Figure 11a, showing that adaptive allocation provides the largest variance reductions and quality gains when particles are limited, while both methods converge as the population grows. The trend for TV-IB, on the other hand, indicates substantial benefits even at higher compute, which is a reflection of the significantly higher search space for audio-visual generation compared to images. Overall, these results support our variance-minimization objective and are consistent with the findings of VASR (Shekhar et al., 2026), which empirically links lower estimator variance to better sample quality.

B.5 Effect of Population Growth Strategy

When the adaptive scheduler increases the particle population, new particles are generated by duplicating existing ones. We compare five growth strategies: *best*, which clones the highest-scoring particle, *weighted*, which samples according to importance weights, *uniform*, which duplicates particles randomly, *spread*, which prioritizes distinct lineages, and *rescue*, which first resamples the population and then expands it by prioritizing particles with fewer offspring. Figure 12 compares their improvement in VR-TA over base generation. *Rescue* achieves the largest improvement (0.099), followed by *weighted* (0.089) and *best* (0.086), while *uniform* performs the worst (0.068). The population schedules remain similar across strategies, suggesting that the performance differences arise primarily from how new particles are selected. These results highlight the benefit of balancing reward-guided selection with lineage preservation during population growth. We therefore use *rescue* in all subsequent experiments.

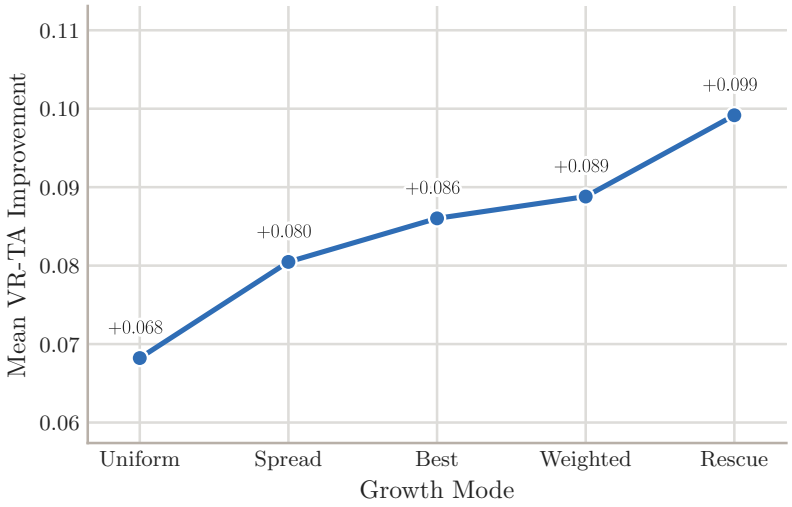
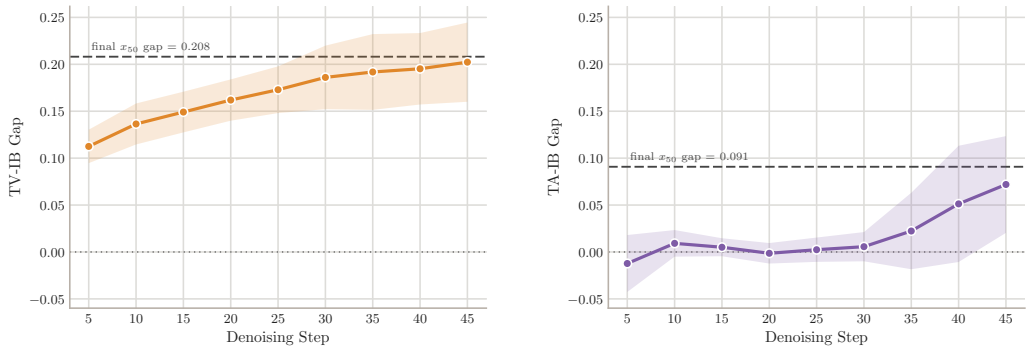


Figure 12: Effect of population growth strategy on VR-TA improvement over base generation. Rescue achieves the largest gain among five strategies, combining reward-guided resampling with lineage preservation during population expansion.Results on JavisDiT++ with 50 prompts and 3 seeds.



(a) TV-IB.

(b) TA-IB.

Figure 13: Reward reliability across denoising checkpoints. We plot the score difference between two dissimilar prompts. Shaded regions show standard deviation across prompts and seeds. TA-IB has little discriminative power early, while TV-IB distinguishes prompts earlier. Both become more discriminative as denoising progresses.