GRADIENT-SPACE EVOLUTION STRATEGIES FOR DIFFUSION MODEL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning aligns diffusion models with learned reward functions, but repeated trajectory evaluation and gradient computation make training expensive. We propose **Gradient-Space Evolution Strategies (GS-ES)**, which organizes alignment around population-based exploration at individual denoising steps. GS-ES perturbs a shared transition mean, evaluates candidate states through gradient-free ODE continuations, and aggregates reward-weighted score-function updates. Selected candidates guide subsequent exploration, while a timestep-dependent budget allocates evaluations according to the remaining rollout length. For Gaussian transitions, we derive an exact identity expressing the local update as a Jacobian pullback of weighted action perturbations. This connects evolution-inspired exploration to gradient-based parameter updates without a reference-policy KL penalty. Experiments with FLUX.1-dev and Wan2.1-T2V-1.3B show higher scores than the evaluated baselines across the listed image and video metrics. On FLUX.1-dev, GS-ES achieves a PickScore of 24.27, compared with 23.33 for DanceGRPO, and reaches the baseline's peak reward in approximately one fifth as many training steps.

1 INTRODUCTION

Diffusion and flow matching models are widely used for image and video generation (Ho et al., 2020; Lipman et al., 2023; Black Forest Labs, 2024; Wan et al., 2025). Building on reinforcement learning from human feedback (Ouyang et al., 2022), recent methods fine-tune these models against learned reward functions. Flow-GRPO (Liu et al., 2025a) and DanceGRPO (Xue et al., 2025) introduce stochastic denoising transitions and formulate generation as a finite-horizon Markov decision process (MDP). Their actions are continuous latent states, even though the trajectory is discretized in time. The resulting policy-gradient framework supports black-box rewards, but its cost depends on how trajectories are sampled, evaluated, and reused for parameter updates.

Two properties of diffusion sampling suggest a more local organization of this computation. First, each transition mean is differentiable with respect to the model parameters. Multiple candidate next states can therefore share one evaluation of that mean, while their reward-weighted update directions can be aggregated before backpropagation. Second, the samplers considered here use a predetermined denoising horizon. The known number of remaining steps exposes a substantial difference in continuation cost between early and late candidate evaluations. These properties motivate exploring several alternatives at a shared intermediate state and allocating the evaluation budget across timesteps.

Local exploration raises three connected design questions: how to generate informative candidates, how to evaluate them without backpropagating through the remaining trajectory, and which candidates should guide subsequent exploration. Recent branching methods already exploit parts of this structure (He et al., 2025; Li et al., 2025b), while mixed ODE–SDE sampling reduces the number of stochastic transitions used for optimization (Li et al., 2025a). Our focus is the combination of structured action perturbations, reward-based population selection, and a direct score-function update at each explored transition.

We introduce **Gradient-Space Evolution Strategies (GS-ES)**, drawing on the perturbation, evaluation, and selection principles of evolution strategies (Rechenberg, 1978; Wierstra et al., 2014; Salimans et al., 2017). For each parent latent state, GS-ES generates candidates around a shared

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.

predicted mean and scores their gradient-free ODE completions. All evaluated candidates contribute to the current update; selected candidates provide states for subsequent exploration. A timestep-dependent budget assigns fewer candidates to expensive, long continuations and more to shorter ones. The training objective uses neither a reference-policy KL penalty nor a clipped likelihood-ratio term.

The connection to gradient space is precise at the level of the local update. For a Gaussian transition, its score function is the model Jacobian transpose applied to a scaled action perturbation. Thus, weighted perturbations can be combined in action space and mapped to parameter gradients with a vector–Jacobian product. This identity motivates the framework's name and explains its computational organization; it does not require evaluating parameter-perturbed models.

The main contributions of this paper are:

1. **A stepwise evolutionary framework for diffusion alignment.** We introduce GS-ES, which combines local candidate generation, gradient-free ODE evaluation, population selection, and adaptive allocation of candidate evaluations. The framework integrates these operations into denoising transitions, using the differentiable transition mean to share computation across exploratory candidates without a reference-policy KL penalty.
2. **An action–gradient correspondence for efficient updates.** We derive an exact Jacobian-pullback identity for the weighted Gaussian score update. It connects structured exploration in action space to update directions in parameter space and explains how candidate information can be aggregated before backpropagation. We also establish the assumptions needed to interpret the expectation of an untransformed reward-weighted update as a local objective's gradient.
3. **Evaluation across image and video generation.** We compare GS-ES with diffusion alignment baselines on FLUX.1-dev and Wan2.1-T2V-1.3B and examine its components through budget, selection, and exploration ablations. The experiments show higher scores on the listed alignment metrics and faster convergence in training steps on the FLUX PickScore task, while the timing comparisons characterize the computational trade-offs.

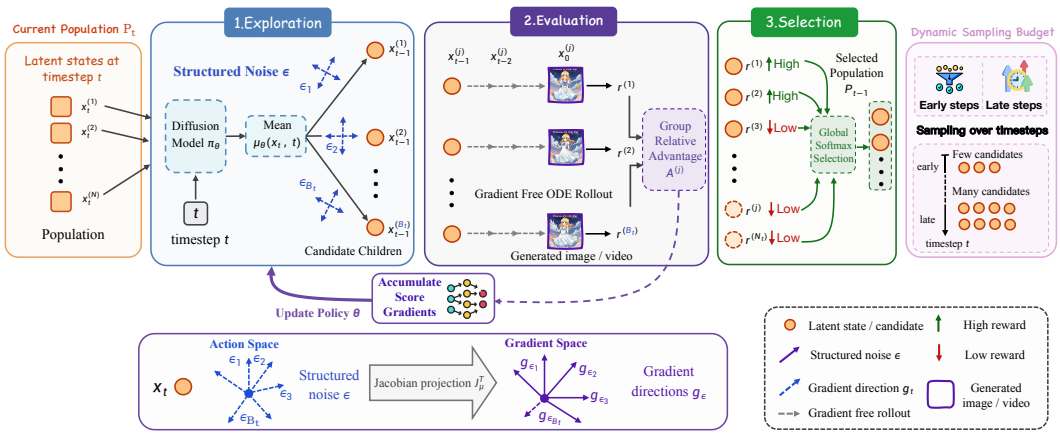

Figure 1: Overview of GS-ES. Candidates share a parent state's transition mean and are evaluated through gradient-free ODE continuations. All candidate rewards contribute to a weighted score update; selected actions guide the next exploration step. A timestep-dependent budget allocates evaluations according to continuation length.

2 Related Work

2.1 Alignment for Diffusion Models

Training-free alignment changes inference without updating model parameters (Song et al., 2023; Tang et al., 2024; Yeh et al., 2025). Preference-based methods learn from ranked pairs, including

diffusion and flow adaptations of direct preference optimization (Wallace et al., 2024; Yang et al., 2024; Liang et al., 2024; Zhang et al., 2025; Liu et al., 2025b). Online methods optimize learned rewards through policy gradients, including DDPO (Black et al., 2023), DPOK (Fan et al., 2023), Flow-GRPO (Liu et al., 2025a), and DanceGRPO (Xue et al., 2025). These methods differ in their sampling and regularization choices; for example, DanceGRPO omits the reference-policy KL penalty by default.

A practical distinction is where reward-guided computation is performed: training-free methods modify individual inference runs, whereas fine-tuning seeks to encode alignment in shared model parameters. GS-ES belongs to the latter setting, while organizing exploration around individual denoising transitions. Candidate evaluation guides both the parameter update and the choice of states for subsequent exploration.

More recent work adapts the rollout and update structure to diffusion generation. TempFlow-GRPO uses trajectory branching and timestep-dependent weighting (He et al., 2025). BranchGRPO combines tree-structured rollouts with process-level advantages and pruning of gradient computations (Li et al., 2025b). MixGRPO uses mixed ODE–SDE sampling (Li et al., 2025a), whereas DiffusionNFT adopts a forward-process fine-tuning objective (Zheng et al., 2026). GS-ES combines structured local exploration with a weighted score update and selection of the states used for subsequent sampling. Its distinction lies in this combination and its computational organization, rather than in being the first method to exploit diffusion structure or omit a reference policy.

2.2 Evolution Strategies and Structured Exploration

Evolution strategies optimize objectives through perturbation and reward-based aggregation or selection (Rechenberg, 1978; Wierstra et al., 2014). In a population-based formulation, mutation generates exploratory candidates, a scalar fitness function evaluates them, and aggregation or selection determines the next search step. This makes ES applicable when objective evaluations are available but derivatives of the objective are not. Gaussian search-distribution methods estimate gradients from evaluations of perturbed candidates (Salimans et al., 2017), and structured random directions can improve the coverage of exploration (Choromanski et al., 2018). Parameter-space noise has also been studied in reinforcement learning (Plappert et al., 2017). Evolutionary algorithms have been applied to inference-time diffusion alignment (Jajal et al., 2025); GS-ES instead updates model parameters through the differentiable transition mean. It evaluates action perturbations and maps their weighted directions through a Jacobian transpose, rather than evaluating one parameter-perturbed model per candidate. For large diffusion models, this organization avoids applying and evaluating a separate full-model parameter perturbation for each exploratory direction. It retains the population-based search structure while using backpropagation through a shared transition prediction to update the generator.

3 Preliminaries

3.1 Diffusion Alignment as Policy Optimization

Let $s_t = (x_t, t, c)$ denote a latent state, denoising step, and prompt. The action $a_t = x_{t-1}$ is the next latent state. Following stochastic formulations of diffusion and flow sampling (Song et al., 2020; Black et al., 2023; Liu et al., 2025a; Xue et al., 2025), we consider the Gaussian transition

$$\pi_\theta(a_t \mid s_t) = \mathcal{N}\big(a_t; \mu_\theta(s_t), \sigma_t^2 \mathbf{I}\big)\,, \tag{1}$$

where μ_θ is the transition mean. The score analysis applies to stochastic steps with $\sigma_t > 0$ independent of θ; it does not apply to a zero-variance deterministic transition. For structured exploration, GS-ES uses this density as a scoring kernel, not necessarily its sampling law.

For the terminal-reward objective $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(x_0, c)]$, the policy-gradient identity is

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t) A^{\pi_\theta}(s_t, a_t)\right], \tag{2}$$

where A^{π_θ} is a valid advantage for the same trajectory policy (Williams, 1992). GRPO-style methods replace this advantage with a group-based estimate and use a clipped likelihood-ratio objective (Shao

et al., 2024). Some variants additionally penalize KL divergence from a reference policy. The behavior policy used in a likelihood ratio and the reference policy used for KL regularization have different roles.

3.2 Gaussian Search in Evolution Strategies

For comparison, Gaussian search in parameter space considers $J_{\mathrm{ES}}(\theta) = \mathbb{E}_{\xi\sim\mathcal{N}(0,\mathbf{I})}[F(\theta+\sigma\xi)]$. Under the usual integrability conditions, an unbiased Monte Carlo estimate of its ordinary gradient is

$$\widehat{\nabla_\theta J_{\mathrm{ES}}} = \frac{1}{N\sigma}\sum_{i=1}^{N} F(\theta+\sigma\xi_i)\xi_i. \tag{3}$$

This score-function estimator is used in Gaussian evolution strategies (Salimans et al., 2017); a natural-gradient update additionally accounts for the search distribution's information geometry (Wierstra et al., 2014). Equations (2) and (3) both weight exploratory directions by objective information, but they sample different objects and generally optimize different objectives.

Structural connection. Policy optimization weights action probes by advantages, whereas parameter-space ES weights model perturbations by fitness. GS-ES uses a differentiable transition mean to combine action-space exploration with parameter updates. The Gaussian score identity, rather than equivalence of the objectives, makes this connection precise.

4 Methods

GS-ES combines a local search procedure with a weighted score-function update. We describe this procedure and establish the identity underlying its gradient-space interpretation.

4.1 Overview

At denoising step t, the current population contains parent states $s_{t,p}$, with $p \in \{1,\dots,P_t\}$. Each parent produces candidate actions by perturbing its predicted transition mean. An ODE continuation and a reward model evaluate each candidate without gradient tracking. The resulting rewards have two roles: they weight the update from all current candidates and determine which states guide subsequent exploration. Only the current transition mean participates in backpropagation. We keep a sampling snapshot $\bar{\theta}$ fixed over the horizon, explore at $t = T,\dots,2$, and take one optimizer step after accumulating these local gradients. The final transition at $t = 1$ is completed by ODE continuation without an additional exploration step. Figure 1 summarizes this procedure.

All evaluated candidates contribute to learning, including those discarded by selection; only selected latents guide subsequent exploration.

4.2 Structured Action-Space Exploration

Candidate $i \in \{1,\dots,B_t\}$ from parent p is

$$a_{t,p,i} = \mu_{\bar{\theta}}(s_{t,p}) + \sigma_t \epsilon_{t,p,i}. \tag{4}$$

All candidates from the same parent share the mean evaluation. The mean includes the stochastic-transition drift correction; Appendix B specifies it and the noise schedule. We compare independent Gaussian noise, antithetic Gaussian pairs $\{\epsilon, -\epsilon\}$, and orthogonal directions with antithetic pairing, drawing on structured exploration in evolution strategies (Salimans et al., 2017; Choromanski et al., 2018). Independent draws provide a reference for local exploration. Antithetic sampling evaluates both sides of a shared direction, making their reward difference available to the update. For the orthogonal variant, QR decomposition supplies the directions and both signs are included. FLUX scales each column by the norm of the corresponding original Gaussian column; Wan uses a fixed radius $\sqrt{d}$, where d is the latent dimension. Distinct underlying directions are orthogonal; the two members of each pair are antiparallel. These designs organize exploration without perturbing model parameters. Their usefulness depends on the local reward landscape and the mapping into parameter updates, so we compare them empirically. Candidates are scored with the Gaussian kernel in Eq. (1);

this does not identify their correlated joint sampling law with independent Gaussian draws. The score identity below holds pointwise, while claims about its expectation require the stated distributional assumptions.

4.3 Gradient-Free Continuation Evaluation

Each candidate is advanced from step $t-1$ to the endpoint using an ODE solver. Endpoint rewards make the candidates comparable without directly interpreting their intermediate latents. Continuation evaluations incur sampling cost but do not retain intermediate activations for backpropagation. Writing this continuation under the frozen snapshot as $\Phi^{\mathrm{ODE}}_{\bar{\theta},t-1\to 0}$, its reward is

$$r_{t,p,i} = R\Big(\Phi^{\mathrm{ODE}}_{\bar{\theta},t-1\to 0}(a_{t,p,i}), c\Big)\,. \tag{5}$$

The continuation and reward evaluation use no gradient tracking. Here R includes the fixed decoder from the endpoint latent to the generated image or video. Within each parent group, we subtract the reward mean and divide by the sample standard deviation plus 10^{-8}, then clip the resulting advantage to $[-5, 5]$. We compare these group-relative weights with rank-based advantages in Appendix C.3. The evaluated candidates and their advantage weights are held fixed when differentiating the local score objective. This separates exploration and reward evaluation from the single-transition gradient computation.

4.4 Sampling Budget and Population Selection

Timestep-dependent budget. Early candidates require longer ODE continuations than later candidates. The schedule therefore restricts expensive early branching and permits more branching near the endpoint. The parameter M_{fix} provides a common scale for comparing these schedules with a fixed candidate count. Under a one-evaluation-per-remaining-step cost model, the continuation length of a candidate at step t is $t-1$. A fixed candidate budget M_{fix} over the $T-1$ exploration steps would incur $\mathcal{C}_{\mathrm{fixed}} = M_{\mathrm{fix}}T(T-1)/2$ continuation evaluations per horizon. Dividing this cost evenly over those steps gives a target continuation budget $M_{\mathrm{fix}}T/2$ per step. To support paired probes, define $\mathcal{E}(n) = \max\{2, n + (n \bmod 2)\}$ for integers n. The target candidate budget is

$$M_t = \mathcal{E}\left(\mathrm{round}\left(\frac{M_{\mathrm{fix}}T}{2\max\{1, t-1\}}\right)\right). \tag{6}$$

This assigns fewer candidates to long continuations and more to short ones. Integer allocation can change the realized candidate count, and decoding, reward evaluation, backpropagation, and batching contribute additional costs. The schedule therefore balances an approximate continuation cost without guaranteeing equal wall-clock time.

Parent and candidate populations. We initialize one parent, $P_T = 1$, and use a target branching factor G to control subsequent parent counts. Given P_t incoming parents, the actual branches per parent and total candidate count are

$$B_t = \mathcal{E}(\lfloor M_t/P_t \rfloor)\,, \qquad N_t = P_t B_t. \tag{7}$$

Thus, G is a target rather than a fixed group size: each parent receives B_t candidates, and the even allocation can make N_t differ from M_t. Each parent's candidates form one group for advantage normalization.

Reward-based selection. After evaluating all parents, we pool their N_t candidates for the prompt. When $t > 2$, the next population has size

$$P_{t-1} = \min\{N_t, \max\{1, \lfloor M_{t-1}/G \rfloor\}\}\,. \tag{8}$$

Softmax selection retains this many distinct candidates without replacement, with weights proportional to $\exp(r_{t,p,i}/\tau)$ across the pool; we use $\tau = 1$. If the requested count exceeds the available candidates, all are retained. Selected latents become detached parent states at step $t-1$. This permits population growth by retaining multiple children of a parent; no additional parent generation is needed. Selection affects subsequent exploration; all evaluated candidates still contribute to the current gradient update. We compare Softmax selection with Greedy, Random, and Tournament alternatives, specified in Appendix C.2. Selection applies reward-based pressure to the pooled population without guaranteeing semantic diversity or constraining the policy's KL divergence. Appendix B.4 gives the complete recursion and update procedure.

4.5 GRADIENT AGGREGATION

Let d be the latent dimension and $A_{t,p,i}$ the detached advantage of candidate i from parent p. The implementation averages the Gaussian log density over latent coordinates and averages candidate contributions within each parent. At the sampling parameters $\bar{\theta}$, its pre-optimizer ascent direction is

$$\begin{aligned} \widehat{g}_{t,p} &= \frac{1}{dB_t} \sum_{i=1}^{B_t} A_{t,p,i} \nabla_\theta \log \pi_\theta(a_{t,p,i} \mid s_{t,p}) \Big|_{\bar{\theta}}, \\ \widehat{g} &= \frac{1}{T-1} \sum_{t=2}^{T} \frac{1}{P_t} \sum_{p=1}^{P_t} \widehat{g}_{t,p}. \end{aligned} \tag{9}$$

The sums include candidates not retained for subsequent exploration. All exploration steps use the same parameter snapshot; gradients are accumulated before one optimizer step. The final transition $t = 1$ is evaluated through ODE continuation. For each parent, candidates share a transition mean and hence a Jacobian. Their weighted action residuals therefore require one vector–Jacobian product per parent. No explicit Jacobian or gradient through an ODE continuation is needed.

Optimization without a reference-policy penalty. Equation (9) uses current candidates without a reference-policy KL penalty or PPO likelihood-ratio clipping. The implemented loss yields this score gradient at $\bar{\theta}$; Appendix A states its normalization explicitly. Gradient norm clipping and AdamW are applied after accumulation. Selection and structured exploration organize the samples; they do not impose a formal bound on parameter changes.

4.6 ACTION–GRADIENT CORRESPONDENCE

The Gaussian density supplies a scoring function for the candidates. With structured exploration, this density need not be their actual sampling law. The following pointwise identity holds for arbitrary candidate directions.

Proposition 1 (Jacobian pullback of a Gaussian score update)**.** *Fix a parent state s, sampling parameters $\bar{\theta}$, a positive variance σ^2 independent of θ, and the Gaussian density in Eq.* (1)*. Let $\mathbf{J}_\mu(s) = \partial \mu_\theta(s)/\partial\theta|_{\bar{\theta}}$. For candidates $a_i = \mu_{\bar{\theta}}(s) + \sigma\epsilon_i$ and scalar weights w_i, holding s, a_i, and w_i fixed during differentiation gives*

$$\sum_i w_i \nabla_\theta \log \pi_\theta(a_i \mid s) \Big|_{\bar{\theta}} = \frac{1}{\sigma} \mathbf{J}_\mu(s)^\top \sum_i w_i \epsilon_i. \tag{10}$$

Proof. At a fixed action, the Gaussian score is $\nabla_\theta \log \pi_\theta(a_i \mid s)|_{\bar{\theta}} = \mathbf{J}_\mu(s)^\top (a_i - \mu_{\bar{\theta}}(s))/\sigma^2 = \mathbf{J}_\mu(s)^\top \epsilon_i/\sigma$. Summing with fixed weights gives Eq. (10). □

In Eq. (9), choosing $w_i = A_{t,p,i}/(dB_t)$ gives $\widehat{g}_{t,p} = \mathbf{J}_\mu(s_{t,p})^\top \sum_i A_{t,p,i}\epsilon_{t,p,i}/(dB_t\sigma_t)$. Thus each probe induces a parameter-space direction $\mathbf{g}_{\epsilon_i} = \mathbf{J}_\mu(s)^\top \epsilon_i$. The transition Jacobian determines the induced directions, and candidate rewards determine their weights. This is the gradient-space interpretation of GS-ES: latent-space exploration supplies directional information that backpropagation maps into parameter updates.

It does not assert that evaluating action perturbations is equivalent to evaluating parameter-perturbed models. Nor does it establish unbiasedness for the global trajectory objective in Eq. (2): structured sampling, group-dependent weights, selected parents, and ODE continuation values change the estimator and its target. Appendix A gives the corresponding local expectation result under explicit assumptions and analyzes which geometric properties survive the Jacobian mapping.

Geometry of the induced directions. The fixed linear pullback preserves antithetic pairing, but orthogonal action probes need not induce orthogonal parameter directions. Their inner product becomes $\epsilon_i^\top \mathbf{J}_\mu \mathbf{J}_\mu^\top \epsilon_j$, as detailed in Appendix A. Action-space coverage therefore does not itself establish lower gradient variance; we assess the exploration choices through the noise ablation.

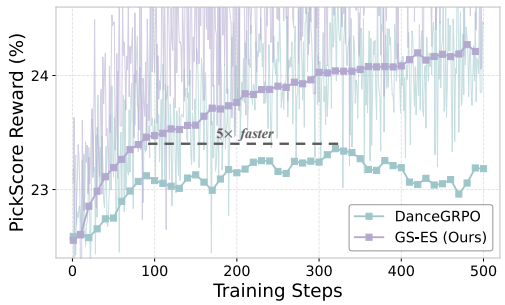

Figure 2: PickScore during FLUX training. Solid curves show test performance; translucent curves show training rewards. The horizontal axis measures training steps.

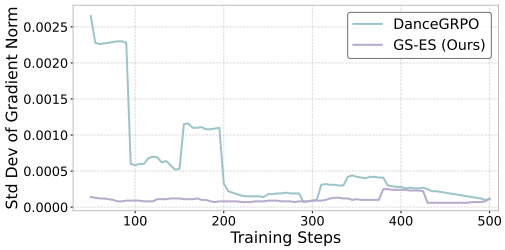

Figure 3: Sliding standard deviation of gradient norms over 50 training steps. This measures temporal variation in update magnitudes, rather than gradient-estimator covariance.

5 Experiments

5.1 Experimental Setup

We evaluate GS-ES on text-to-image (T2I) generation with FLUX.1-dev (Black Forest Labs, 2024) and text-to-video (T2V) generation with Wan2.1-T2V-1.3B (Wan et al., 2025). The comparisons include Flow-GRPO (Liu et al., 2025a), DanceGRPO (Xue et al., 2025), BranchGRPO (Li et al., 2025b), and DiffusionNFT (Zheng et al., 2026). Within each task, all methods start from the same base model.

Rewards and evaluation. For T2I, we train separate models with PickScore (Kirstain et al., 2023), HPSv2 (Wu et al., 2023), and HPSv3 (Ma et al., 2025). We evaluate PickScore on 500 test prompts from Pick-a-Pic v1 and HPSv2/HPSv3 on 400 test prompts from HPDv2. ImageReward (Xu et al., 2023) and UnifiedReward (Wang et al., 2025) serve as held-out reward evaluators; their reported scores are averaged over the models trained with the three training rewards. For T2V, prompts come from VidProM (Wang & Yang, 2024), with 400 prompts reserved for testing. We conduct separate training runs using PickScore, HPSv2, and the text-alignment score of VideoReward from VideoAlign (Liu et al., 2025b), denoted VideoAlign in Table 1. We additionally evaluate video quality and semantic alignment with VBench (Huang et al., 2023). These additional evaluators test transfer across evaluation criteria, without alone establishing generalization to a shifted prompt distribution.

Configuration. All experiments use eight NVIDIA H800 GPUs. We follow the DanceGRPO baseline in our comparisons for the optimizer learning rate, weight decay, and gradient-norm clipping. The default GS-ES configuration uses the timestep-dependent budget with $M_{\text{fix}} = 16$, orthogonal action-space exploration, and Softmax selection. Appendix C.1 clarifies the shared optimization settings and the scope of the forward-count reporting.

5.2 Main Results

Alignment scores. Table 1 reports the highest score for GS-ES in each listed image and video metric. For T2I, GS-ES obtains 24.27 PickScore, 39.75 HPSv2, and 15.56 HPSv3, compared with 23.33, 37.81, and 15.00 for DanceGRPO. Its ImageReward and UnifiedReward scores also improve, suggesting transfer to evaluators not used for the corresponding training objectives. For T2V, GS-ES obtains 20.31 PickScore, 28.79 HPSv2, and 1.46 VideoAlign, alongside higher VBench scores than the evaluated baselines. We report point estimates throughout.

Training cost and convergence. GS-ES shifts computation from gradient-enabled predictions to gradient-free candidate evaluation. The timing results in Table 1 depend on the task: a T2I training step takes 720 s for GS-ES and 740 s for DanceGRPO, whereas a T2V step takes 390 s and 360 s, respectively. Thus the reduction in gradient-enabled evaluations does not imply a uniform reduction in time per step. On the FLUX PickScore task, Figure 2 shows that GS-ES reaches DanceGRPO's peak reward in approximately one fifth as many training steps. The final PickScore gains over the base

Table 1: Alignment scores and training cost for FLUX.1-dev and Wan2.1-T2V-1.3B. ImageReward and UnifiedReward are held-out evaluators, averaged over the three image training objectives; VBench provides additional video evaluation. Grad / Free denotes the reported gradient-enabled / gradient-free forward counts per training step, and Time is seconds per step. **Bold**: highest score; underline: second highest.

Text-to-Image (Base Model: FLUX.1-dev)

Method	In-domain Alignment Metrics			Held-out Reward Avg.		Efficiency (per step)	
	PickScore ↑	HPSv2 ↑	HPSv3 ↑	Image Reward ↑	Unified Reward ↑	Grad / Free	Time (s)
FLUX.1-dev	22.58	29.55	13.91	1.112	3.370	–	–
Flow-GRPO	23.40	37.33	15.02	1.526	3.382	432 / 768	740
DanceGRPO	23.33	37.81	15.00	1.539	3.379	432 / 768	740
BranchGRPO	23.57	38.12	15.13	1.563	3.388	360 / 980	620
DiffusionNFT	<u>23.91</u>	<u>38.27</u>	<u>15.18</u>	<u>1.614</u>	<u>3.401</u>	180 / 1120	680
GS-ES	**24.27**	**39.75**	**15.56**	**1.627**	**3.406**	26 / 1466	720

Text-to-Video (Base Model: Wan2.1-T2V-1.3B)

Method	In-domain Alignment Metrics			Additional Evaluation on VBench			Efficiency (per step)	
	PickScore ↑	HPSv2 ↑	VideoAlign ↑	Quality ↑	Semantic ↑	Total ↑	Grad / Free	Time (s)
Wan2.1	18.92	16.45	0.72	83.27	73.07	81.23	–	–
Flow-GRPO	19.61	24.34	0.99	83.45	73.86	81.53	576 / 960	360
DanceGRPO	19.35	19.93	0.93	83.32	73.29	81.31	576 / 960	360
BranchGRPO	19.78	25.63	1.08	83.61	74.18	81.72	480 / 1240	340
DiffusionNFT	<u>19.84</u>	<u>25.92</u>	<u>1.12</u>	<u>83.74</u>	<u>74.63</u>	<u>81.91</u>	240 / 1680	370
GS-ES	**20.31**	**28.79**	**1.46**	**83.96**	**75.42**	**82.26**	37 / 2130	390

model are 1.69 for GS-ES and 0.75 for DanceGRPO, a ratio of approximately 2.25. The convergence comparison concerns training steps; the table separately reports measured time per step.

Gradient-norm fluctuations. Figure 3 shows the standard deviation of gradient norms over a sliding window of 50 training steps. GS-ES exhibits smaller temporal fluctuations in this diagnostic than DanceGRPO. Because gradient directions can vary even when their norms remain constant, this plot does not measure the covariance of the gradient estimator or prove unbiasedness. It provides a descriptive view of the observed training dynamics, complementary to the reward curves.

Qualitative comparisons. Figure 4 in Appendix C.4 presents generated samples from the base model, Flow-GRPO, DanceGRPO, GS-ES, BranchGRPO, and DiffusionNFT under the displayed prompts. The examples cover different objects, attributes, and generation scenarios, allowing inspection of visual fidelity and prompt adherence alongside the numerical scores. They complement the reward-model evaluations by showing the kinds of output differences observed across methods. We use these selected examples to illustrate the results, while Table 1 reports performance over the evaluation prompt sets.

5.3 Ablation Studies

We ablate the budget schedule, selection operator, and exploration noise on FLUX.1-dev. Unless otherwise stated, results are differences from the default configuration: timestep-dependent budget with $M_{\text{fix}} = 16$, Softmax selection, and orthogonal exploration.

Budget allocation. Table 2 compares candidate-budget schedules. This comparison tests whether the placement of candidate evaluations matters in addition to their overall cost. With a fixed candidate count, early denoising steps require longer ODE continuations for each reward evaluation. The timestep-dependent schedule instead allocates more candidates to later steps, where each continuation is shorter, making local exploration less expensive on a per-candidate basis. At the same reported

Table 2: Candidate-budget schedule (vs. Default $M_{\text{fix}} = 16$).

Schedule (Setting)	Δ PickScore	Δ HPSv3	Time
Fixed $M = 16$	-0.51	-0.33	1.00×
Varying $M_{\text{fix}} = 8$	-0.42	-0.24	0.67×
Varying $M_{\text{fix}} = 24$	+0.03	+0.08	1.28×

Table 3: Selection operator (vs. Softmax).

Selection Strategy	Δ PickScore	Δ HPSv3
Greedy	-0.25	-0.23
Random Selection	-0.06	-0.15
Tournament Selection	-0.17	-0.21

Table 4: Exploration noise (vs. Orthogonal).

Noise Type	Δ PickScore	Δ HPSv3
Antithetic Sampling	-0.08	-0.19
Random Gaussian	-0.15	-0.36

relative step time, using fixed $M = 16$ lowers PickScore by 0.51 and HPSv3 by 0.33. Increasing M_{fix} to 24 improves these scores by 0.03 and 0.08 while increasing time per step to $1.28\times$ the default. Reducing it to 8 lowers the scores by 0.42 and 0.24 and reduces time per step to $0.67\times$. Among these tested settings, the default offers a useful trade-off between reward and cost.

Selection. Table 3 compares the selection operators. Greedy selection lowers PickScore by 0.25 relative to Softmax, while Random and Tournament selection lower it by 0.06 and 0.17. The corresponding HPSv3 differences follow the same ordering. Softmax selection gives higher-reward candidates a greater probability of continuing while still allowing alternatives to survive. We hypothesize that this balance explains its advantage over Greedy selection, which can concentrate continuation on a narrow set of candidates, and Random selection, whose survival decisions ignore reward. The ablation compares final reward scores and does not directly measure population diversity. Selection should also be distinguished from gradient aggregation: candidates that do not survive can still contribute reward-weighted directions to the current update.

Exploration noise. Table 4 compares orthogonal exploration with antithetic and independent Gaussian alternatives. Relative to orthogonal directions, antithetic sampling lowers PickScore/HPSv3 by 0.08/0.19, and independent Gaussian sampling lowers them by 0.15/0.36. These alternatives probe complementary ways to organize a finite set of local perturbations: antithetic pairs balance opposite directions, whereas orthogonal directions reduce directional overlap in action space. The results favor the latter design in this experiment. The score identity connects these sampled directions to parameter updates, but the reward comparisons do not by themselves identify the mechanism responsible for the observed differences.

Advantage weighting. We also compare group-relative weights with an ES-inspired rank-based alternative, which maps rewards to ranks normalized to $[-0.5, 0.5]$ before gradient aggregation. On the FLUX PickScore task, this alternative lowers PickScore by 0.17 relative to the default. We hypothesize that reward magnitudes provide useful information for distinguishing locally similar candidates, whereas ranking retains only their ordering. This result favors group-relative weighting in the tested setting; further details are given in Appendix C.3.

6 Conclusion

GS-ES organizes diffusion alignment around local candidate exploration, gradient-free ODE evaluation, reward-based selection, and weighted score updates. Its theoretical core is an exact Gaussian score identity: reward-weighted action perturbations can be aggregated and mapped to parameter gradients through the transition mean's Jacobian transpose. The timestep-dependent budget allocates more candidate evaluations to shorter continuations. Experiments show higher image and video alignment scores than the evaluated baselines, with faster convergence in training steps on the FLUX PickScore task. Together, these results support population-based local exploration as a practical approach to diffusion model alignment. This perspective also motivates investigating how other evolutionary search mechanisms can be combined with differentiable diffusion transitions.

AI USE STATEMENT

LLMs were used for language polishing, technical review, and code debugging, and the authors take responsibility for the final manuscript.

ETHICS STATEMENT

This study uses publicly available datasets and does not involve private, sensitive, or personally identifiable data. Its experiments optimize image and video generators against learned reward functions. Scores from these evaluators characterize the tested criteria and should not be interpreted as a comprehensive assessment of human preferences or deployment safety.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Black Forest Labs. FLUX.1-dev: Model card. Hugging Face, 2024. URL https://huggingface.co/black-forest-labs/FLUX.1-dev.

Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pp. 970–978. PMLR, 2018.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. TempFlow-GRPO: When timing matters for GRPO in flow models. *arXiv preprint arXiv:2508.04324*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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, 2023. URL https://arxiv.org/abs/2311.17982.

Purvish Jajal, Nick John Eliopoulos, Benjamin Shiue-Hal Chou, George K. Thiruvathukal, James C. Davis, and Yung-Hsiang Lu. Inference-time alignment of diffusion models via evolutionary algorithms. *arXiv preprint arXiv:2506.00299*, 2025. URL https://arxiv.org/abs/2506.00299.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE. *arXiv preprint arXiv:2507.21802*, 2025a.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. BranchGRPO: Stable and efficient GRPO with structured branching in diffusion models, 2025b. URL https://arxiv.org/abs/2509.06040.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Ji Li, and Liang Zheng. Step-aware preference optimization: Aligning preference with denoising performance at each step. *arXiv preprint arXiv:2406.04314*, 2024. URL https://arxiv.org/abs/2406.04314v1.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. *arXiv preprint arXiv:2505.05470*, 2025a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025b.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15086–15095, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

Ingo Rechenberg. Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie: Workshop, Hannover, 29. Sept.–1. Okt. 1977*, pp. 83–114. Springer, 1978.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pp. 32483–32498. PMLR, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Tuning-free alignment of diffusion models with direct noise optimization. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

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

Wenhao Wang and Yi Yang. VidProM: A million-scale real prompt-gallery dataset for text-to-video diffusion models. *Advances in Neural Information Processing Systems*, 37:65618–65642, 2024.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025. URL https://arxiv.org/abs/2503.05236.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(27):949–980, 2014.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. DanceGRPO: Unleashing GRPO on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8941–8951, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Using_Human_Feedback_to_Fine-tune_Diffusion_Models_without_Any_Reward_CVPR_2024_paper.html.

Po-Hung Yeh, Kuang-Huei Lee, and Jun-Cheng Chen. Training-free diffusion model alignment with sampling demons, 2025. URL https://arxiv.org/abs/2410.05760.

Tao Zhang, Cheng Da, Kun Ding, Huan Yang, Kun Jin, Yan Li, Tingting Gao, Di Zhang, Shiming Xiang, and Chunhong Pan. Diffusion model as a noise-aware latent reward model for step-level preference optimization. *arXiv preprint arXiv:2502.01051*, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process, 2026. URL https://arxiv.org/abs/2509.16117.

A Theoretical Details

A.1 Implemented score objective and normalization

All $T-1$ exploration steps use one parameter snapshot $\bar{\theta}$. Parent states, sampled candidates, and advantage weights are detached; the ODE continuation does not receive gradients. Write $\ell_\theta(a \mid s) = d^{-1} \log \pi_\theta(a \mid s)$ for the coordinate-averaged Gaussian log density used in the implementation. For the collected candidates, define the fixed-data surrogate

$$\begin{aligned} \widehat{\mathcal{S}}(\theta) &= \frac{1}{T-1} \sum_{t=2}^{T} \frac{1}{P_t B_t} \sum_{p=1}^{P_t} \sum_{i=1}^{B_t} A_{t,p,i} \rho_{t,p,i}(\theta), \\ \rho_{t,p,i}(\theta) &= \exp(\ell_\theta(a_{t,p,i} \mid s_{t,p}) - \ell_{\bar{\theta}}(a_{t,p,i} \mid s_{t,p})) \,. \end{aligned} \tag{11}$$

The implementation minimizes the negative of this expression at $\theta = \bar{\theta}$, using a detached copy of the current log score in the denominator. The exponential is then one, but its derivative is $\nabla_\theta \ell_\theta$. Consequently $\nabla_\theta \widehat{\mathcal{S}}(\bar{\theta}) = \widehat{g}$ in Eq. (9). This is a local derivative calculation, not a likelihood-ratio correction between separately evaluated policies. The $1/d$ factor comes from averaging over latent coordinates; $1/B_t$, $1/P_t$, and $1/(T-1)$ come from candidate, parent, and exploration-step averaging, respectively. Because $P_t B_t$ varies, these weights cannot all be absorbed into one common constant across timesteps. After gradient accumulation, gradient norm clipping and AdamW produce the parameter update; $\widehat{g}$ denotes the pre-optimizer ascent direction, not the final parameter displacement.

A.2 Local objective and expectation identity

Proposition 1 is an identity for a finite weighted candidate set. We now state the corresponding expectation result. Fix a timestep t, evaluator parameters $\bar{\theta}$, and a parent-state distribution q_t. Let $\Phi_{\bar{\theta},t-1\to 0}$ denote the deterministic ODE continuation, and define

$$Q_{\bar{\theta},t}(a,s) = R\big(\Phi_{\bar{\theta},t-1\to 0}(a;s), c(s)\big)\,, \qquad \mathcal{L}_t(\theta) = \mathbb{E}_{s\sim q_t} \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[Q_{\bar{\theta},t}(a,s)]. \tag{12}$$

Here $c(s)$ is the prompt contained in s. Both q_t and the evaluator are held fixed when differentiating $\mathcal{L}_t$; the evaluator need not be differentiable.

Corollary (local expectation identity). Suppose $\sigma_t > 0$ is independent of θ, the transition mean is differentiable, and the expectations below are finite and permit differentiation under the integral. For any action-independent baseline $b(s)$,

$$\nabla_\theta \mathcal{L}_t(\theta) = \mathbb{E}_{s\sim q_t}\left[\frac{\mathbf{J}_\theta(s)^\top}{\sigma_t} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\big(Q_{\bar{\theta},t}(\mu_\theta(s)+\sigma_t\epsilon, s) - b(s)\big)\epsilon\right]\right], \tag{13}$$

where $\mathbf{J}_\theta(s) = \partial\mu_\theta(s)/\partial\theta$. For a fixed parent, an empirical average of these raw-reward terms is unbiased whenever every candidate direction has marginal distribution $\mathcal{N}(0,I)$; the directions may be dependent. Coordinate-averaged log scores multiply this result by $1/d$.

Proof. Since the evaluator and q_t are fixed, differentiating under the integral gives $\nabla_\theta \mathcal{L}_t = \mathbb{E}[Q_{\bar{\theta},t}(a,s)\nabla_\theta \log \pi_\theta(a \mid s)]$. The conditional expectation of the score is zero, so subtracting $b(s)$ leaves this expression unchanged. Substituting the Gaussian score in Proposition 1 gives Eq. (13). Linearity of expectation proves the empirical-average statement using only the individual Gaussian marginals. □

This result makes the local optimization target explicit. At a sampling parameter value $\theta = \bar{\theta}$, it describes the untransformed reward-weighted update with a frozen continuation. Selection determines which parents are evaluated later, so q_t need not equal the occupancy of the unselected stochastic policy. Likewise, an ODE continuation value need not equal that policy's stochastic continuation value. These distinctions separate the local objective from the complete trajectory objective in Eq. (2).

A.3 GROUP-RELATIVE WEIGHTS

The practical update uses group-dependent advantage weights. Proposition 1 applies to these weights when detached, whereas the corollary concerns raw rewards or action-independent baselines. Their expectation properties differ even in a simple case. For n independent actions from the same fixed-parent Gaussian transition, let $R_i = Q_{\bar{\theta},t}(a_i, s)$, $\bar{R} = n^{-1}\sum_i R_i$, and $\ell_s(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[Q_{\bar{\theta},t}(a, s)]$. Under the same differentiation assumptions,

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(R_i - \bar{R})\nabla_\theta \log \pi_\theta(a_i \mid s)\right] = \left(1 - \frac{1}{n}\right)\nabla_\theta \ell_s(\theta). \tag{14}$$

The diagonal terms contribute n copies of $\nabla_\theta \ell_s$, while the cross terms vanish by independence and the zero-mean score property. Subtracting the group mean therefore removes one fraction $1/n$ of the expectation. For dependent candidates, the cross terms require separate analysis. For a complete antithetic set, the unweighted scores sum to zero at $\bar{\theta}$, so subtracting a shared group mean cancels pointwise before any nonlinear clipping. This cancellation does not make the group-standardized update an unbiased raw-reward estimator. Division by a sample standard deviation and advantage clipping introduce further transformations. We use these operations as practical reward scaling; the raw-reward unbiasedness result is not a guarantee for the transformed estimator.

A.4 CONNECTION TO EVOLUTION STRATEGIES

Equation (13) combines zeroth-order estimation in action space with a first-order Jacobian pullback into parameter space. This is the precise connection underlying GS-ES. Parameter-space ES instead evaluates an objective at perturbed parameters. Substituting $\xi = \mathbf{J}_\theta^\top \epsilon$ into a parameter perturbation does not identify these evaluations: for a fixed state,

$$\mu_{\theta+\nu\mathbf{J}_\theta^\top \epsilon} = \mu_\theta + \nu \mathbf{J}_\theta \mathbf{J}_\theta^\top \epsilon + o(\nu). \tag{15}$$

The induced action displacement generally differs from $\sigma_t \epsilon$. Moreover, NES defines a natural-gradient update for a search distribution (Wierstra et al., 2014); the score pullback alone does not specify that update. GS-ES adopts fitness-weighted exploration and selection while sharing the model computation across action perturbations.

A.5 GEOMETRY OF STRUCTURED EXPLORATION

The pullback is linear at a fixed parent and parameter value. Thus an antithetic pair remains antithetic, and its raw-reward average, using the joint Gaussian log density before the $1/d$ implementation scaling, has the form

$$\widehat{g}_{\text{pair}} = \frac{Q(\mu + \sigma\epsilon) - Q(\mu - \sigma\epsilon)}{2\sigma}\mathbf{J}^\top \epsilon. \tag{16}$$

Zero-mean directions also remain zero-mean under this fixed linear map. Orthogonality requires more care: for $g_i = \mathbf{J}^\top \epsilon_i$,

$$g_i^\top g_j = \epsilon_i^\top \mathbf{J}\mathbf{J}^\top \epsilon_j, \qquad \mathrm{Cov}(g_i) = \mathbf{J}^\top \mathrm{Cov}(\epsilon_i)\mathbf{J}. \tag{17}$$

Orthogonal action directions therefore need not remain orthogonal in parameter space. Their action-space coverage and the variance of the final weighted update are distinct properties.

A.6 SAMPLING LAWS OF THE STRUCTURED PROBES

Let $H = B_t/2 \le d$ and draw a matrix $Z = [z_1, \ldots, z_H] \in \mathbb{R}^{d\times H}$ with independent standard Gaussian entries. A reduced QR factorization supplies orthonormal columns $u_1, \ldots, u_H$. The image implementation uses $\epsilon_j = \|z_j\|_2 u_j$ and includes $-\epsilon_j$; the video implementation instead uses $\epsilon_j = \sqrt{d}\, u_j$ and its negative. Thus the positive-index half is orthogonal in action space, while the full candidate set contains opposite pairs and is not pairwise orthogonal. The two implementations use different radial laws. The image radii have χ_d marginals, whereas the video radii are constant. QR sign conventions and the dependence of the constructed directions also matter when identifying a sampling law; orthogonality or a radius distribution alone does not establish that each indexed direction is standard Gaussian.

In particular, fixed-radius probes are not Gaussian samples. For these probes, $\nabla_\theta \log \pi_\theta(a \mid s)$ in Proposition 1 is the gradient of the Gaussian scoring density evaluated at a sampled point; it is not the score of the actual fixed-radius sampling distribution. The pointwise identity remains exact because it requires no distributional assumption on ϵ_i. The expectation corollary has the additional Gaussian-marginal assumption and is not asserted for every structured-noise variant or for the group-normalized, clipped estimator.

B Transition and Implementation Details

B.1 Transition mean and noise schedule

We distinguish the discrete denoising index t from the continuous flow time u_t. The shifted grid is $u_t = h(t/T)$, where $h(z) = \kappa z/[1 + (\kappa - 1)z]$ and $\kappa = 3$. Thus $u_T = 1$, $u_0 = 0$, and $\Delta_t = u_t - u_{t-1} > 0$. Let $v_\theta(x_t, u_t, c)$ denote the model's velocity prediction, including its fixed conditioning settings. Its time encoding uses $\lfloor 1000u_t \rfloor/1000$ for FLUX and $\lfloor 1000u_t \rfloor$ for Wan, while the integration coefficients use the unrounded grid. The implementation forms an endpoint prediction and a drift-correction term,

$$\widehat{x}_0 = x_t - u_t v_\theta(x_t, u_t, c),$$
$$h_\theta(s_t) = -\frac{x_t - (1 - u_t)\widehat{x}_0}{u_t^2 + 10^{-6}}.$$

The stochastic candidate mean and standard deviation in Eq. (4) are

$$\mu_\theta(s_t) = x_t - \Delta_t v_\theta(x_t, u_t, c) + \tfrac{1}{2}\eta_t^2 \Delta_t h_\theta(s_t), \qquad \sigma_t = \eta_t \sqrt{\Delta_t}, \tag{18}$$

with $\eta_t = \eta_0[1 - 0.9(T - t)/T]^\gamma$, $\eta_0 = 5$, and $\gamma = 1.5$. The correction in Eq. (18) is part of the differentiable mean; its derivative is included in $\mathbf{J}_\mu$. Its inclusion specifies the implemented sampler and does not establish exact marginal preservation under discretization or structured noise. ODE continuations instead use $x_{t-1} = x_t - \Delta_t v_{\bar{\theta}}(x_t, u_t, c)$, without stochastic noise or the correction term. The final $t = 1$ transition is evaluated only within these continuations and contributes no local score gradient.

B.2 Sampling and gradient boundaries

One optimization iteration keeps the generator parameters at $\bar{\theta}$ while constructing and evaluating the entire population sequence. The initial latent is Gaussian, with one parent per prompt. For each parent, the velocity prediction and corrected transition mean retain gradients; candidate sampling, decoder and reward evaluation, ODE continuation, and survivor selection do not. Selected latent states are detached before becoming parents at the next timestep. Advantages are also detached, including their sample mean, sample standard deviation, and clipping operation. Consequently, gradients do not pass through parent selection, earlier transitions, or the evaluator.

The implementation evaluates the dimension-averaged Gaussian log density $\ell_\theta = d^{-1} \log \pi_\theta$ and uses the score surrogate

$$L_{t,p}(\theta) = -\frac{1}{B_t} \sum_{i=1}^{B_t} A_{t,p,i} \exp(\ell_\theta(a_{t,p,i} \mid s_{t,p}) - \mathrm{sg}[\ell_\theta(a_{t,p,i} \mid s_{t,p})]), \tag{19}$$

where sg stops differentiation. The exponential equals one in the forward pass, and its first derivative is $\nabla_\theta \ell_\theta$. Thus minimizing this surrogate, weighted by $1/[(T - 1)P_t]$ for each parent, implements Eq. (9). It is not a behavior-policy likelihood ratio or a higher-order representation of the log-score objective. After all parents and timesteps have contributed, the accumulated gradient is clipped and AdamW updates the parameters once.

B.3 Evaluation work and auxiliary diagnostics

The actual number of candidate continuation predictions per prompt is $\sum_{t=2}^{T} N_t(t - 1)$, while gradient-enabled parent predictions number $\sum_{t=2}^{T} P_t$. These counts distinguish sample evaluations

from batched device calls and exclude checkpoint recomputation, decoding, reward models, and auxiliary diagnostics. The supplied implementation additionally evaluates an initial full ODE trajectory for monitoring; its reward is not subtracted from the group-relative advantages. The FLUX implementation also retains a frozen reference model for diagnostics, although no reference-policy KL penalty enters the training loss. These diagnostics can incur computation and memory costs even though they do not contribute to the update.

B.4 Complete Population Recursion and Update Procedure

We describe one optimization horizon for one prompt c. Let T be the training denoising horizon, M_{fix} the budget scale, and G the target branching factor. For an integer n, define $\mathcal{E}(n) = \max\{2, n + (n \bmod 2)\}$, which increases an odd integer to the next even integer and enforces a minimum of two. The implementation first rounds the continuous budget to the nearest integer (ties to even), then applies $\mathcal{E}$. The actual candidate count can differ from the target budget because every parent receives the same even number of branches.

1. **Initialize.** Set the sampling snapshot $\bar{\theta} \leftarrow \theta$ and clear the accumulated parameter gradients. Draw one initial latent x_T, set $P_T = 1$, and form the initial parent state $s_{T,1} = (x_T, T, c)$. Hold $\bar{\theta}$ fixed until the horizon is complete.
2. **Allocate branches.** For $t = T, T-1, \ldots, 2$, compute the target budget M_t using Eq. (6). Given the incoming parent count P_t, set $B_t = \mathcal{E}(\lfloor M_t / P_t \rfloor)$ and $N_t = P_t B_t$.
3. **Explore and evaluate each parent.** For each $p = 1, \ldots, P_t$, compute $\mu_{\bar{\theta}}(s_{t,p})$ once with gradient tracking enabled. Generate B_t structured perturbations and form the candidates in Eq. (4) without tracking gradients through sampling. For each candidate, run the deterministic ODE continuation through the remaining $t-1$ transitions, decode the endpoint, and evaluate its reward. All continuation and reward computations use $\bar{\theta}$ without gradient tracking.
4. **Accumulate the local gradient.** Compute the reward mean and sample standard deviation separately within each parent group. Center and scale each reward using that group's statistics, add 10^{-8} to the standard deviation, and clip the resulting advantage to $[-5, 5]$. Holding the parent, candidate actions, and advantage weights fixed, evaluate their Gaussian log scores through the shared transition mean and backpropagate the normalized contribution in Eq. (9). Every candidate contributes, including candidates that will not survive selection. Free the local computation graph after this backward pass; retain detached candidates and rewards for selection.
5. **Choose the next population.** Pool all N_t candidates from the current parents and set

$$K_{t-1} = \max\{1, \lfloor M_{t-1}/G \rfloor\}, \qquad P_{t-1} = \min\{N_t, K_{t-1}\}.$$

   If $K_{t-1} \geq N_t$, retain the full pool. Otherwise, select K_{t-1} distinct candidates using the chosen selection operator. Softmax selection draws without replacement with weights proportional to $\exp(r/\tau)$ over the pooled rewards; the experiments use $\tau = 1$. Use the selected candidate latents as the detached parents for step $t-1$. If $t > 2$, repeat allocation, evaluation, gradient accumulation, and selection at the next exploration step. The implementation also performs this selection at $t = 2$, using the bookkeeping value M_1 defined by Eq. (6); the resulting parents are not expanded further.
6. **Update once.** After processing $t = 2$, clip the accumulated parameter gradient using the configured gradient-norm threshold and take one optimizer step. The final transition at $t = 1$ has already been evaluated by the ODE continuations; it generates no additional exploratory group or score-gradient contribution. The final survivor selection changes neither the evaluated rewards nor the accumulated gradient for this horizon.

This recursion allows the number of parents to increase by retaining several children from the current pool. It requires neither artificial duplication of a parent nor an unspecified replenishment step. Group boundaries are used for advantage normalization; survival is decided across the full pool for the prompt, so one parent can contribute multiple survivors and another none.

C Additional Experimental Details

C.1 Experimental Settings

GS-ES follows the DanceGRPO baseline in our comparisons for the AdamW learning rate, weight decay, and maximum gradient norm. Its candidate generation, timestep-dependent budget, population selection, and gradient aggregation follow Section 4 and Appendix B.4. The default uses orthogonal exploration and Softmax selection; the budget ablation settings are specified in Table 2. The shared optimizer settings do not specify the denoising horizon, batch size, or population schedule.

The per-prompt work expressions in Appendix B describe the sample-level cost of the algorithm. A run-level reconciliation of the reported forward counts in Table 1 with the experiment configurations remains outstanding.

C.2 Selection Operators

Let $\mathcal{C} = \{1, \dots, N\}$ index a pool of evaluated candidates with rewards r_i. The pool contains all $N_t = P_t B_t$ candidates for the current prompt and timestep, across parent states. For the without-replacement operators below, $K = \min\{N_t, \max(1, \lfloor M_{t-1}/G \rfloor)\}$ is the number of retained indices. When $K = N_t$, all candidates are retained. The selected actions become parent states for subsequent exploration.

Softmax (default). Starting from $\mathcal{P} = \mathcal{C}$, sample the next index with probability

$$p(i \mid \mathcal{P}) = \frac{\exp((r_i - r_{\max})/\tau)}{\sum_{j \in \mathcal{P}} \exp((r_j - r_{\max})/\tau)}, \qquad i \in \mathcal{P}, \tag{20}$$

where $r_{\max} = \max_{j \in \mathcal{P}} r_j$ and $\tau = 1.0$. Subtracting $r_{\max}$ leaves the probabilities unchanged and improves numerical stability. Remove the sampled index from $\mathcal{P}$ and repeat until K indices have been selected. These are conditional draw probabilities, not the final inclusion probabilities of the selected subset.

Greedy and Random. Greedy retains the K highest-reward candidates from the full pool; Random draws K distinct indices uniformly without using their rewards.

Tournament. For each draw, sample $\min\{3, |\mathcal{P}|\}$ distinct candidates uniformly from the current available pool $\mathcal{P}$. Retain its highest-reward candidate, remove that index from $\mathcal{P}$, and repeat until K candidates have been retained. Only the winner is removed; unselected contestants remain available for later draws. Index uniqueness does not imply distinct latent states or semantic diversity.

C.3 Rank-Based Advantage

We compare group-relative advantage weights with rank-based weights, which replace reward magnitudes with ranks normalized to $[-0.5, 0.5]$. On the FLUX PickScore task, rank-based weighting lowers PickScore by 0.17 relative to the default group-relative weighting. One possible explanation is that reward differences contain useful information that is discarded by ranking; this experiment alone does not establish that mechanism.

C.4 Qualitative Comparisons

Figure 4 compares selected outputs from the base model, the baselines, and GS-ES under the displayed prompts. The examples illustrate visual attributes and prompt adherence and complement the quantitative evaluation in Table 1.

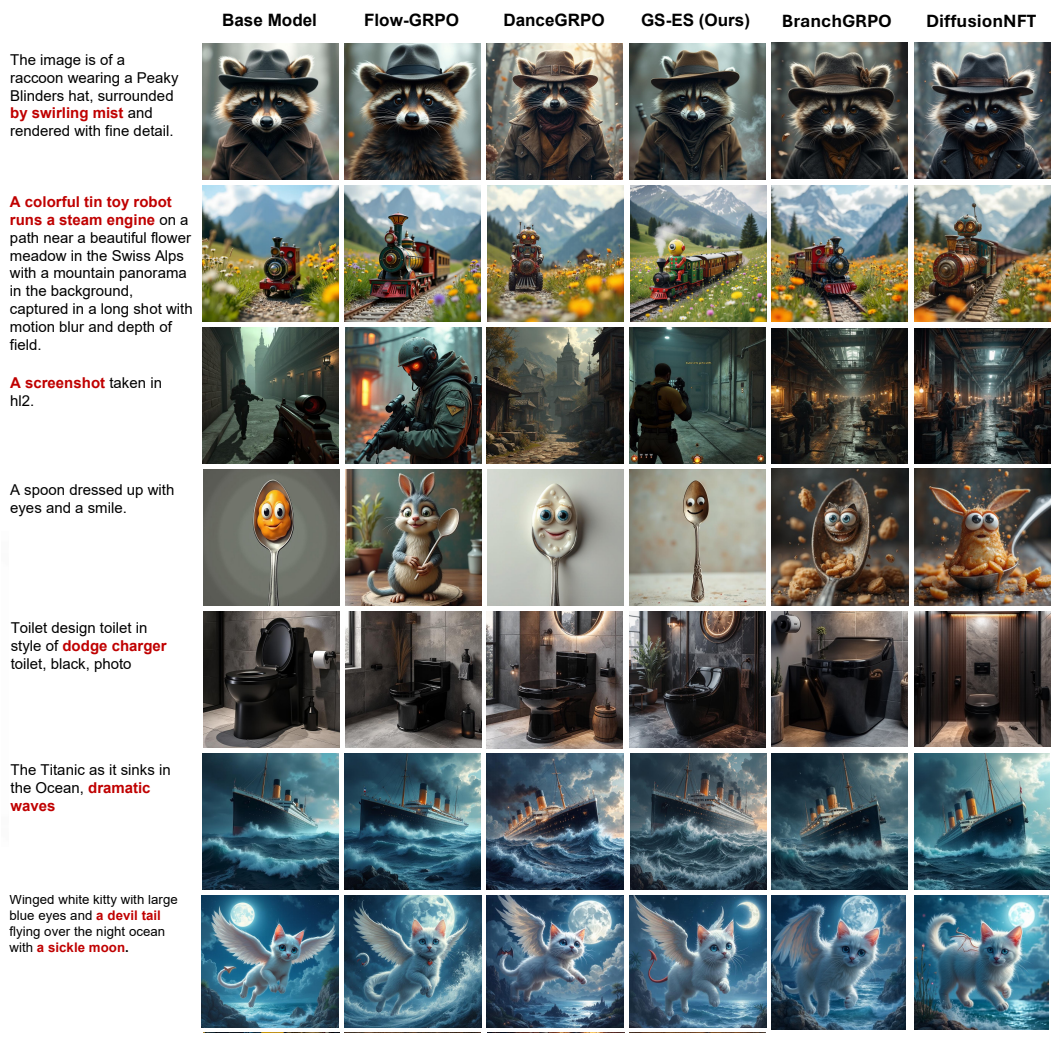

Figure 4: Selected qualitative comparisons under the displayed prompts. Columns show the base model, Flow-GRPO, DanceGRPO, GS-ES, BranchGRPO, and DiffusionNFT. These examples illustrate output characteristics and complement the quantitative comparisons.