EXPLORING THE DESIGN SPACE OF REWARD BACKPROPAGATION FOR FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct reward backpropagation aligns text-to-image flow matching models sample-efficiently, but practical methods fix different surrogate backward trajectories, leaving unclear which choices drive performance. Full-rollout differentiation requires substantial activation memory and can amplify gradients through chained Jacobians. Surrogates reduce these costs but introduce fidelity issues: detached methods query rewards on model-implied clean estimates, while connector methods such as LeapAlign use derivatives of single-velocity interval approximations. We propose FlowBP, a framework that treats the backward trajectory as the design object and exposes four axes: reward-model input, active set, integration weights, and bridge coupling. Prior direct-gradient methods occupy specific settings in this space. Transplanting endpoint reconstruction into ReFL and DRTune improves all five metrics on `FLUX.1-dev` with active sets and optimizer unchanged. Three instantiations evaluate the sampled image, eliminating connector residuals through Euler reconstruction (FlowBP-Sparse, FlowBP-Bridge) or targeting smaller residuals through multi-support quadrature (FlowBP-Lagrange). They retain activations for only a few velocity calls and at most one nested state Jacobian. Across `SD3.5-M`, `FLUX.1-dev`, and `FLUX.2-Klein-base`, they improve on prior direct-gradient baselines in preference metrics and overall compositional prompt following. Balancing alignment gains, training time, and tuning reliability, we recommend FlowBP-Sparse with $K = 3$ active calls as the default recipe.

1 INTRODUCTION

Pretrained text-to-image flow matching models (Liu et al., 2023; Lipman et al., 2023; Esser et al., 2024; Labs, 2024; Black Forest Labs, 2026) produce high-quality images but can benefit from preference alignment. Two post-training paradigms address this gap. *Policy-gradient methods* (Black et al., 2024; Liu et al., 2025; Xue et al., 2025) make the sampler stochastic and optimize learned rewards through likelihood-style estimators. *Direct reward backpropagation* (Xu et al., 2023; Clark et al., 2024; Wu et al., 2024; Liang et al., 2026) instead differentiates the reward through the inference trajectory. The latter is sample-efficient and avoids stochastic policy-gradient variance, but full-rollout differentiation requires substantial activation memory and chained Jacobians can amplify gradients (Clark et al., 2024). The methods studied here therefore replace the full backward trajectory with a short or sparse *surrogate*, whose design remains open.

Once the backward path departs from the rollout, a third difficulty appears: surrogate fidelity. The surrogate can disagree with the trajectory it replaces in two distinct ways. First, the reward may be queried on a quantity the sampler never produced: ReFL (Xu et al., 2023) and DRTune (Wu et al., 2024) evaluate the reward on a model-implied one-step clean estimate taken from an intermediate state, which may lie off the natural-image manifold the reward model was trained on. Second, the backward path may be pinned to the rollout by a straight-through connector, as in LeapAlign (Liang et al., 2026), leaving a measurable discrepancy between the cached forward state and the surrogate state whose derivative is used. DRaFT-LV (Clark et al., 2024) mixes the two cases: its direct final-step target reproduces the sampled image, whereas its re-noised final-step predictions do not. These are questions of *fidelity*, and each prior method fixes one configuration by construction, so it remains unclear which choice drives performance and which configurations no method occupies.

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.

We propose FlowBP, a surrogate-trajectory framework that treats the backward trajectory itself as the design object. It caches a no-gradient rollout and builds a sparse backward graph from selected re-forwarded velocities. Four explicit choices determine this graph: the reward-model input, active velocities, integration weights, and bridge coupling. This common design space makes prior methods directly comparable and exposes new combinations (Table 1).

Our contributions are as follows:

- We introduce FlowBP, a unified surrogate-trajectory framework that separates reward-model input, active velocities, integration weights, and bridge coupling, and maps ReFL, DRaFT-LV, DRTune, and LeapAlign into a common design space.
- We instantiate three complementary variants that evaluate the reward on the sampled endpoint while retaining activations for only K re-forwarded calls. FlowBP-Sparse uses exact Euler endpoint reconstruction, FlowBP-Bridge adds one coupled reconstruction bridge, and FlowBP-Lagrange uses multi-support Lagrange connectors to target smaller residuals. Their backward graphs contain at most one state Jacobian.
- We derive the full Euler-rollout Jacobian and the exact Jacobians of the surrogate graphs. Effective coverage, connector residual, and nested depth describe complementary structural properties, while forward-error and worst-case amplification bounds characterize their behavior without claiming recovery of the full-backpropagation gradient.
- Across three backbones, the best FlowBP variant outperforms every prior direct-gradient baseline on 14 of 15 backbone–metric pairs, and FlowBP-Lagrange raises GenEval overall from 63.25 to 69.88. Ablations probe all four axes, while endpoint-reconstruction transplants improve all five metrics for both ReFL and DRTune on `FLUX.1-dev`. These ablations yield a practical default: FlowBP-Sparse with $K = 3$.

2 Related Work

Preference alignment for diffusion and flow models. Recent work explores diverse post-training strategies for aligning visual generative models with human preferences. Policy-gradient and GRPO-style methods adapt RL objectives to diffusion and flow-model sampling trajectories (Black et al., 2024; Fan et al., 2023; Liu et al., 2025; Xue et al., 2025; Li et al., 2025), while preference-optimization methods learn from paired, step-wise, dense, or self-generated preference signals (Wallace et al., 2024; Yang et al., 2024a; Liang et al., 2025; Yuan et al., 2024; Yang et al., 2024b; Zhang et al., 2025a). Related likelihood- or score-based objectives further align diffusion models through score matching, discriminative optimization, or stochastic optimal control (Zhu et al., 2025; Zheng et al., 2025; Domingo-Enrich et al., 2025). Other flow or diffusion alignment methods connect online reward optimization with the model's pretraining or forward noising process (Zheng et al., 2026; Xue et al., 2026). Many of these methods accommodate non-differentiable rewards or pairwise preferences and optimize stochastic trajectory objectives, preference likelihoods, control objectives, or likelihood-style surrogates rather than directly differentiating the reward through the sampler. Our work studies this complementary direct-gradient regime.

Direct reward backpropagation. Direct-gradient methods exploit the differentiability of diffusion or flow samplers to optimize reward scores without likelihood estimation (Prabhudesai et al., 2023; Xu et al., 2023; Clark et al., 2024; Wu et al., 2024; Zhang et al., 2025b; Shen et al., 2025; Liang et al., 2026). Existing approaches backpropagate rewards through short late-stage denoising paths, low-variance final-step objectives, detached intermediate inputs, composition-aware feedback, full-trajectory preference signals, or learned leap trajectories. These methods demonstrate the effectiveness of differentiable rewards, but each fixes a particular choice of reward input, active denoising steps, trajectory approximation, and nested-gradient scale. Our surrogate-trajectory view makes these choices explicit and uses them as design axes for the methods studied in this paper.

3 Preliminaries

Diffusion and flow models. Diffusion models (Ho et al., 2020; Song et al., 2021) and flow matching (Lipman et al., 2023; Liu et al., 2023) both admit deterministic continuous-time sampling under suitable parameterizations. For Gaussian interpolation paths, a velocity field $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ describes the

evolution of $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$ between clean data $\boldsymbol{x}_0 \sim p_{\text{data}}$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, conditioned on a prompt $\boldsymbol{c}$. Sampling integrates the ODE $\mathrm{d}\boldsymbol{x}_t/\mathrm{d}t = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$; the learned field and its relation to other prediction parameterizations depend on the training formulation and schedule (Salimans & Ho, 2022). We study the rectified-flow setting (Liu et al., 2023; Esser et al., 2024), with $\alpha_t = 1 - t$ and $\sigma_t = t$, and analyze the discrete Euler sampler used in our experiments.

In this setting, the straight-line interpolation $\boldsymbol{x}_t = (1-t)\,\boldsymbol{x}_0 + t\,\boldsymbol{\epsilon}$ has target velocity $\boldsymbol{v} = \mathrm{d}\boldsymbol{x}_t/\mathrm{d}t = \boldsymbol{\epsilon} - \boldsymbol{x}_0$, and the network is trained against this target by minimizing

$$\mathcal{L}_{\text{FM}}(\theta) \;=\; \mathbb{E}_{t,\,\boldsymbol{x}_0,\,\boldsymbol{\epsilon}}\Big[\,\big\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) - (\boldsymbol{\epsilon} - \boldsymbol{x}_0)\big\|_2^2\,\Big]. \tag{1}$$

To generate samples, we discretize reverse time into an N-step rollout with schedule $\sigma_N = 1 > \sigma_{N-1} > \cdots > \sigma_0 = 0$, where $\sigma_i \in [0, 1]$ is the time (equivalently, noise level) at step i; for example, a uniform discretization sets $\sigma_i = i/N$. Starting from initial noise $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the Euler update

$$\boldsymbol{x}_{i-1} \;=\; \boldsymbol{x}_i \;-\; (\sigma_i - \sigma_{i-1})\,\boldsymbol{v}_i, \qquad \boldsymbol{v}_i := \boldsymbol{v}_\theta(\boldsymbol{x}_i, \sigma_i, \boldsymbol{c}), \qquad i = N, \ldots, 1, \tag{2}$$

produces a trajectory $\{\boldsymbol{x}_i\}_{i=N}^{0}$ that terminates at the clean sample $\boldsymbol{x}_0$. Since the rollout is fully differentiable, the gradient of a differentiable reward $r(\boldsymbol{x}_0, \boldsymbol{c})$ can be backpropagated through any subset of these N steps to update the parameters θ (Xu et al., 2023; Clark et al., 2024; Wu et al., 2024; Liang et al., 2026); in latent-space models, r includes the fixed image decoder and reward preprocessing.

Straight-through connectors. A common primitive used by trajectory-level reward methods such as LeapAlign (Liang et al., 2026) is the *straight-through connector*: at any rollout index i, given a differentiable surrogate estimate $\hat{\boldsymbol{x}}_i$, the in-place pin

$$\boldsymbol{x}_i \;=\; \hat{\boldsymbol{x}}_i + \mathrm{sg}(\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i), \tag{3}$$

keeps the forward value of $\boldsymbol{x}_i$ equal to its cached value while rerouting the gradient through $\hat{\boldsymbol{x}}_i$, so that $\partial \boldsymbol{x}_i/\partial\theta = \partial \hat{\boldsymbol{x}}_i/\partial\theta$. Following LeapAlign, we continue to use the symbol $\boldsymbol{x}_i$ for the post-connector latent. We refer to $d_i = \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|_2$ as the *connector residual* at index i: it does not affect the forward value but quantifies the discrepancy between the forward and backward paths at the connection point.

4 Method

We align a pretrained model by maximizing $\mathbb{E}[r(\boldsymbol{x}_0, \boldsymbol{c})]$. Full-rollout differentiation is the reference update; our methods replace its backward graph.

4.1 Setup and Challenges

Full-rollout backpropagation has two known costs, while replacing it with a surrogate introduces a third (Clark et al., 2024; Wu et al., 2024; Liang et al., 2026).

Activation memory. Retaining activations for all N velocity calls is expensive at modern model scale.

Gradient explosion. The Euler chain multiplies state Jacobians $\mathbf{I} - (\sigma_i - \sigma_{i-1})\frac{\partial \boldsymbol{v}_\theta(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_i}$. The product admits an exponential worst-case bound in their accumulated magnitude and can destabilize early-step updates; this does not imply exponential growth in N itself.

Surrogate infidelity. Detached short surrogates may query the reward on a model-implied estimate $\hat{\boldsymbol{x}}_0$ rather than the sampled image. Connector-based methods query the cached $\boldsymbol{x}_0$ but compose its reward gradient with $\partial \hat{\boldsymbol{x}}_0/\partial\theta$. Their residual $d_0 = \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\|_2$ records a forward discrepancy, not a derivative bound; empirically, large residuals coincide with instability (Figure 4).

4.2 Unified Surrogate-Trajectory Framework

We first run Equation (2) without autograd and cache $\{\boldsymbol{x}_i\}_{i=0}^{N}$ and $\{(\boldsymbol{v}_i, \sigma_i)\}_{i=1}^{N}$. A second pass then re-forwards only selected velocities: its forward states are attached to the cached rollout, while its

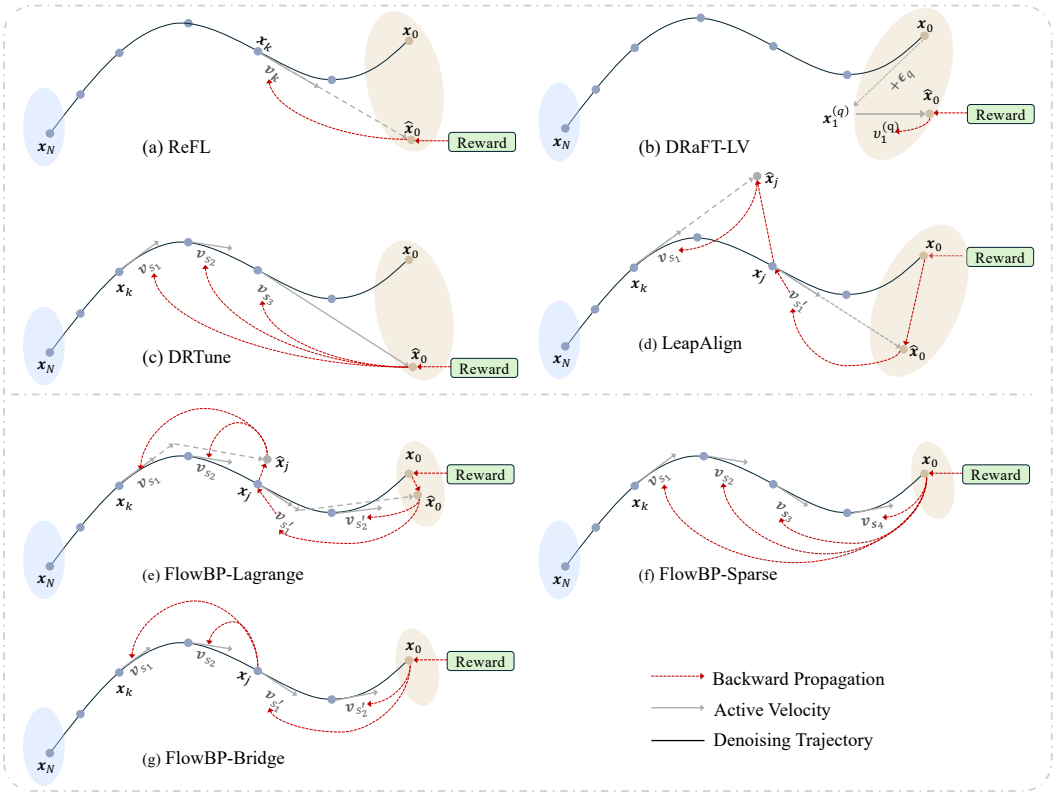


Figure 1: Surrogate backward graphs. Highlighted velocity calls retain gradients; other trajectory quantities are detached. Reward input, supports, weights, and bridge settings are summarized in Table 1.

sparse backward graph replaces full-trajectory backpropagation. Figure 1 visualizes this construction and the corresponding graphs of prior methods.

Cached segments and active velocities. The surrogate uses one segment ending at $\boldsymbol{x}_0$, or two segments $\boldsymbol{x}_k \to \boldsymbol{x}_j \to \boldsymbol{x}_0$ with $N \geq k > j \geq 0$. Reconstruction methods set $k = N$. Only the active set $\mathcal{A} = \mathcal{A}_{\text{pre}} \cup \mathcal{A}_{\text{post}}$ is re-forwarded with autograd, where $\mathcal{A}_{\text{pre}} \subseteq (j, k]$, $\mathcal{A}_{\text{post}} \subseteq (0, j]$, and $K = |\mathcal{A}|$. An interior bridge requires $j \in \mathcal{A}_{\text{post}}$; $j = 0$ denotes no bridge. For a segment rule with base weights $\{w_i\}$, each cached velocity is replaced by

$$\tilde{\boldsymbol{v}}_i = \begin{cases} \boldsymbol{v}_\theta(\alpha \boldsymbol{x}_j + (1-\alpha)\,\mathrm{sg}(\boldsymbol{x}_j), \sigma_j, \boldsymbol{c})\,, & i = j \in \mathcal{A}, \\ \boldsymbol{v}_\theta(\mathrm{sg}(\boldsymbol{x}_i), \sigma_i, \boldsymbol{c})\,, & i \in \mathcal{A},\ i \neq j, \\ \mathrm{sg}(\boldsymbol{v}_i), & i \notin \mathcal{A}. \end{cases} \tag{4}$$

Thus every non-bridge active call has a detached input, while the bridge call at j retains an α-scaled dependence on the split latent. Stop-gradient rescaling multiplies support i's backward signal by ρ_i while preserving its forward value:

$$\tilde{\boldsymbol{v}}_i^{\rho} = \mathrm{sg}(\tilde{\boldsymbol{v}}_i) + \rho_i\big(\tilde{\boldsymbol{v}}_i - \mathrm{sg}(\tilde{\boldsymbol{v}}_i)\big), \qquad \beta_i := \rho_i w_i \tag{5}$$

Here β_i is the effective backward weight.

Attaching the forward graph. For any interval $s \to t$, the surrogate connects to the cached state in one of two ways:

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_s - \sum_{i \in \mathcal{S}_{s\to t}} w_i \tilde{\boldsymbol{v}}_i^{\rho}, \qquad \boldsymbol{x}_t = \hat{\boldsymbol{x}}_t + \mathrm{sg}(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t) \qquad \text{(connector)}, \tag{6}$$

$$\boldsymbol{x}_t = \boldsymbol{x}_s - \sum_{i=t+1}^{s} h_i \tilde{\boldsymbol{v}}_i^{\rho}, \qquad h_i = \sigma_i - \sigma_{i-1} \qquad \text{(Euler reconstruction)}. \tag{7}$$

The connector uses the cached value and the derivative of $\hat{\boldsymbol{x}}_t$; $d_t = \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2$ measures their forward discrepancy. Euler reconstruction instead reproduces the cached discrete state exactly in exact arithmetic, assuming deterministic re-evaluation at unchanged parameters, because each $\tilde{\boldsymbol{v}}_i$ has forward value $\boldsymbol{v}_i$. At the endpoint, both constructions can make the reward read the sampled $\boldsymbol{x}_0$, whereas short detached methods may query a model-implied estimate $\hat{\boldsymbol{x}}_0$. At an interior split, the bridge call in Equation (4) is evaluated after $\boldsymbol{x}_j$ has been attached: $\alpha = 0$ detaches the bridge input, while $\alpha \neq 0$ retains one state Jacobian.

One backward graph. For these cached-rollout graphs, either attachment rule gives

$$\frac{\partial \boldsymbol{x}_0}{\partial \theta} = \underbrace{-\sum_{i \in \mathcal{A}} \beta_i \frac{\partial \boldsymbol{v}_\theta(\boldsymbol{x}_i)}{\partial \theta}}_{\text{direct terms}} + \underbrace{\alpha \beta_j \frac{\partial \boldsymbol{v}_\theta(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j} \sum_{i \in \mathcal{A}_{\text{pre}}} \beta_i \frac{\partial \boldsymbol{v}_\theta(\boldsymbol{x}_i)}{\partial \theta}}_{\text{nested term}}. \tag{8}$$

The first term updates active calls independently; the second is present only for a coupled bridge and contains one state Jacobian regardless of N. Here network partial derivatives hold the input fixed, and $\beta_j = 0$ without a bridge. Table 1 summarizes each graph by its *effective coverage* $W_{\mathcal{A}} = \sum_{i \in \mathcal{A}} |\beta_i|$, the total absolute backward weight ($W_{\mathcal{A}} = 1$ for full backpropagation), and its *nested depth*, the largest number of state Jacobians on a backward path (0 without a coupled bridge, 1 with one, and $N - 1$ for full backpropagation); neither is a gradient-fidelity score (Appendix A.3). Retained activations scale with K rather than N. The amplification bound is affine in one state-Jacobian norm, replacing the full chain's exponential bound (Proposition 3).

Design axes. The construction separates four choices: the *reward-model input*, the *active set*, the base *integration weights* w_i, and the *bridge coupling* (split construction and α). These characterize forward agreement and backward-graph structure rather than defining a scalar distance to full backpropagation. Sample weighting and optimizer settings also affect the final update (Appendix B).

4.3 Method Instantiations

Prior methods and our variants can now be compared through these axes. Our three instantiations all keep the reward on the sampled $\boldsymbol{x}_0$: FlowBP-Sparse uses one exact Euler reconstruction, FlowBP-Bridge splits that reconstruction around a coupled bridge, and FlowBP-Lagrange uses two multi-support connector segments. Table 1 provides the roadmap before the detailed descriptions.

Table 1: Settings of the four design axes, with $\beta_i = \rho_i w_i$ and $W_{\mathcal{A}} = \sum_{i \in \mathcal{A}} |\beta_i|$. Residuals and depth do not measure gradient fidelity. DRaFT-LV's n re-noised targets $\boldsymbol{z}_{1:n}$ require separate active calls; its zero error refers only to its direct target ($q = 0$). h and Δ denote segment length and maximum Euler step; full BP retains J_i from Equation (13).

Method	RM input	Active calls	Base weight	Bridge	$W_{\mathcal{A}}$	Fwd. error	Depth
ReFL	$\hat{\boldsymbol{x}}_0^{(\tau)}$	$\{\tau\}$	σ_τ	none	σ_τ	$O(h^2)$	0
DRaFT-LV	$\boldsymbol{x}_0, \boldsymbol{z}_{1:n}$	final/target	σ_1/target	none	σ_1/target	0 ($q = 0$)	0
DRTune	$\hat{\boldsymbol{x}}_0$	train steps $\geq t_{\min}$	h_i; close $\sigma_{t_{\min}}$	none	$\sum_{\mathcal{A}} \lvert\beta_i\rvert$	$O(h^2)$	0
LeapAlign	$\boldsymbol{x}_0$	$\{k, j\}$	$\sigma_k - \sigma_j,\ \sigma_j$	conn., α	σ_k	$O(h^2)$	1
FlowBP-Sparse	$\boldsymbol{x}_0$	K	h_i	none	$\sum_{\mathcal{A}} \lvert\beta_i\rvert$	0	0
FlowBP-Bridge	$\boldsymbol{x}_0$	K, split at j	h_i	recon., α	$\sum_{\mathcal{A}} \lvert\beta_i\rvert$	0	1
FlowBP-Lagrange	$\boldsymbol{x}_0$	$K \leq 2M$	w_i^{L}	conn., α	$\sum_{\mathcal{A}} \lvert\beta_i\rvert$	$O(h^{M+1} + h\Delta)$	1
Full BP	$\boldsymbol{x}_0$	all N	h_i	full chain	1	0	$N - 1$

Prior methods. ReFL (Xu et al., 2023) uses one active tail predictor, while DRTune (Wu et al., 2024) uses sparse rollout calls and an independently sampled closing predictor; both query a model-implied clean estimate. DRaFT-LV (Clark et al., 2024) uses a direct final-step target, which reproduces the cached $\boldsymbol{x}_0$, plus fresh active final-step predictions at detached re-noised inputs, which do not. Each of these targets has nested depth zero. LeapAlign (Liang et al., 2026) instead connects both segments with one active anchor each, so the reward reads $\boldsymbol{x}_0$ and the backward graph contains one bridge Jacobian. Its long single-velocity intervals can have large residuals d_j and d_0, whose association with training instability is studied in Figure 4. Detailed update graphs are given in Appendix C.

FlowBP-Sparse. FlowBP-Sparse sets $j = 0$, replays the complete Euler sum from $\boldsymbol{x}_N$, and uses $\beta_i = \rho_i h_i$ on K active calls:

$$\boldsymbol{x}_0 = \boldsymbol{x}_N - \sum_{i=1}^{N} h_i \tilde{\boldsymbol{v}}_i^{\rho}. \tag{9}$$

All cached velocities participate in the forward sum, so the endpoint is exact; only $\mathcal{A}$ participates backward, giving nested depth zero and $O(K)$ retained activations.

FlowBP-Bridge. FlowBP-Bridge chooses $j \in \{1, \ldots, N-1\}$, applies Euler reconstruction separately on $(N \to j)$ and $(j \to 0)$, and substitutes $\beta_i = \rho_i h_i$ into Equation (8). Both the bridge and endpoint match the cached rollout in the forward pass. For this fixed split graph, $\alpha = 0$ removes the nested term, while $\alpha \neq 0$ adds one-Jacobian cross-segment coupling; it is not identical to FlowBP-Sparse unless the active sets and backward scales also coincide; see Appendix A.6 for the derivation.

FlowBP-Lagrange. FlowBP-Lagrange keeps LeapAlign's two connector segments but uses M Lagrange supports per segment. With supports $\mathcal{S}_{\text{pre}} \subseteq (j, k]$ and $\mathcal{S}_{\text{post}} \subseteq (0, j]$, it substitutes $\beta_i = \rho_i w_i^{\text{L}}$ into Equation (8), where ρ_i attenuates the non-anchor supports. The one-support, unscaled case recovers LeapAlign's structural gradient; additional supports target smaller connector residuals without changing the nested depth of one. Weights, support scaling, and the complete derivation are given in Appendix A.7.

5 Experiments

5.1 Setup

We evaluate our methods on three text-to-image flow-matching backbones spanning different model families and capacities: `SD3.5-M` (Esser et al., 2024), `FLUX.1-dev` (Labs, 2024), and `FLUX.2-Klein-base` (9B) (Black Forest Labs, 2026). For each backbone, we report the base model and four reward-gradient post-training baselines: ReFL (Xu et al., 2023), DRaFT-LV (Clark et al., 2024), DRTune (Wu et al., 2024), and LeapAlign (Liang et al., 2026). We implement these baselines following their official papers. On `FLUX.1-dev` we additionally report MixGRPO (Li et al., 2025), a policy-gradient method, as a reference point outside the direct-gradient family.

Following MixGRPO (Li et al., 2025), we evaluate all methods on the HPDv2 test split, which contains 400 held-out prompts. We use HPSv2.1 (Wu et al., 2023), PickScore (Kirstain et al., 2023), and ImageReward (Xu et al., 2023) to measure human preference alignment, and UnifiedReward-Alignment (UR-Align) and UnifiedReward-IQ (UR-IQ) (Wang et al., 2025) to assess image-text alignment and overall image quality. For each metric, we report the average score over the 400 prompts and the improvement over the corresponding base model; every number is the mean over three random seeds (`SD3.5-M` standard deviations in Table 7). Since post-training uses HPSv2.1 as the differentiable reward, we treat HPSv2.1 as the in-domain metric and the remaining reward models as out-of-domain evaluators.

5.2 Implementation Details

For our variants, we follow the LeapAlign training protocol and optimize the DiT parameters against HPSv2.1 (Wu et al., 2023) as the differentiable reward model, using a hinge threshold of $\lambda = 0.55$ and training prompts sampled from HPDv2. Online rollouts use 25 sampling steps at 512×512 resolution, and each checkpoint is evaluated with 50-step sampling under the same resolution and classifier-free guidance (Ho & Salimans, 2022) scale as training. Baselines share these common settings and differ only in their method-specific hyperparameters. The full optimizer settings, per-backbone training configuration, and method-specific hyperparameters are given in Appendix B. The comparison runs use $K = 4$ for both reconstruction variants; the budget ablations below motivate our default recommendation of FlowBP-Sparse with $K = 3$.

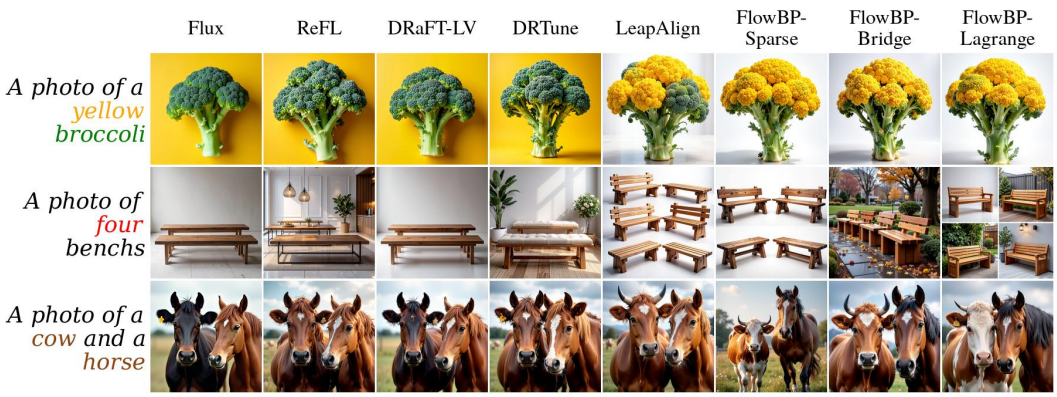


Figure 2: GenEval examples on `FLUX.1-dev`: rows share a prompt and columns compare methods under object, attribute, counting, and spatial constraints.

Table 2: Preference scores across three backbones. Δ is the gain over the base model; best/second-best scores are **bold**/underlined. Blue rows are our variants. †MixGRPO is a policy-gradient reference.

Method	In-Domain		Out-of-Domain							
	HPSv2.1		PickScore		ImageReward		UR-Align		UR-IQ	
	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)
SD3.5-M										
`SD3.5-M`	0.2881	—	22.5004	—	0.9504	—	3.3935	—	3.6376	—
ReFL	0.3872	+0.0991	23.4021	+0.9017	1.3799	+0.4295	3.4779	+0.0844	3.9464	+0.3088
DRaFT-LV	0.3818	+0.0937	23.3506	+0.8502	1.3673	+0.4169	3.5072	+0.1137	3.9245	+0.2869
DRTune	0.3870	+0.0989	23.2882	+0.7878	1.3794	+0.4290	3.4562	+0.0627	3.8869	+0.2493
LeapAlign	0.3924	+0.1043	23.3142	+0.8138	1.4425	+0.4921	3.4854	+0.0919	3.9030	+0.2654
FlowBP-Sparse	0.3939	+0.1058	23.4104	+0.9100	1.4299	+0.4795	3.4774	+0.0839	**3.9690**	**+0.3314**
FlowBP-Bridge	0.3966	+0.1085	23.4130	+0.9126	1.4386	+0.4882	3.4712	+0.0777	3.9481	+0.3105
FlowBP-Lagrange	**0.3998**	**+0.1117**	**23.5040**	**+1.0036**	**1.4538**	**+0.5034**	**3.5127**	**+0.1192**	3.9680	+0.3304
FLUX.1-dev 12B										
`FLUX.1-dev`	0.3016	—	22.5940	—	1.0094	—	3.3168	—	3.7551	—
ReFL	0.3930	+0.0914	23.6886	+1.0946	1.3916	+0.3822	3.5005	+0.1837	4.0293	+0.2742
DRaFT-LV	0.3900	+0.0884	23.6290	+1.0350	1.3910	+0.3816	3.5264	+0.2096	4.0259	+0.2708
DRTune	0.3920	+0.0904	23.6704	+1.0764	1.4273	+0.4179	3.5403	+0.2235	4.0697	+0.3146
LeapAlign	0.4063	+0.1047	23.6158	+1.0218	1.4373	+0.4279	3.5290	+0.2122	4.0584	+0.3033
MixGRPO†	0.3686	+0.0670	23.5262	+0.9322	**1.6083**	**+0.5989**	3.4563	+0.1395	3.6182	−0.1369
FlowBP-Sparse	0.4081	+0.1065	23.7458	+1.1518	1.4742	+0.4648	3.5408	+0.2240	4.0897	+0.3346
FlowBP-Bridge	0.4119	+0.1103	**23.7952**	**+1.2012**	1.5156	+0.5062	3.5681	+0.2513	**4.1098**	**+0.3547**
FlowBP-Lagrange	**0.4134**	**+0.1118**	23.7796	+1.1856	1.4906	+0.4812	**3.5848**	**+0.2680**	4.0950	+0.3399
FLUX.2-Klein-base 9B										
`FLUX.2-Klein-base`	0.2873	—	22.3158	—	1.1405	—	3.6392	—	3.7982	—
ReFL	0.4272	+0.1399	23.6002	+1.2844	1.6245	+0.4840	3.6817	+0.0425	4.0565	+0.2583
DRaFT-LV	0.4233	+0.1360	23.6288	+1.3130	1.6262	+0.4857	3.6495	+0.0103	3.9319	+0.1337
DRTune	0.4122	+0.1249	23.5716	+1.2558	1.6264	+0.4859	3.6638	+0.0246	4.0589	+0.2607
LeapAlign	0.4246	+0.1373	23.4884	+1.1726	1.6553	+0.5148	**3.6926**	**+0.0534**	4.0625	+0.2643
FlowBP-Sparse	0.4384	+0.1511	**23.8446**	**+1.5288**	**1.6649**	**+0.5244**	3.6720	+0.0328	**4.0787**	**+0.2805**
FlowBP-Bridge	**0.4387**	**+0.1514**	23.7718	+1.4560	1.6491	+0.5086	3.6775	+0.0383	4.0720	+0.2738
FlowBP-Lagrange	0.4321	+0.1448	23.6938	+1.3780	1.6376	+0.4971	3.6705	+0.0313	4.0652	+0.2670

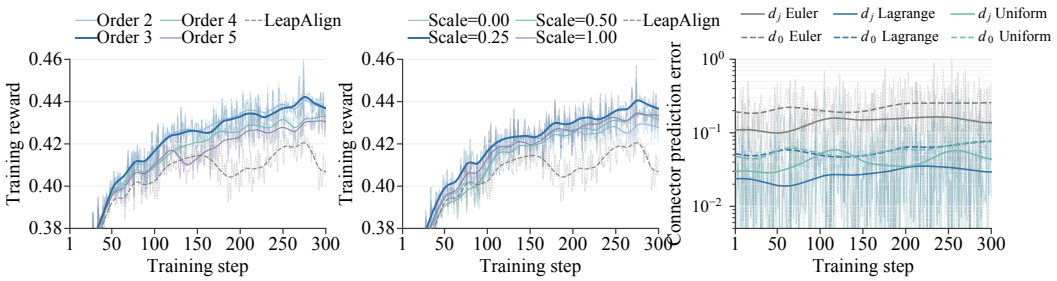


Figure 3: FlowBP-Lagrange ablations on `FLUX.1-dev`. Left: quadrature order. Middle: gradient-support scale. Right: Euler, Lagrange, and uniform connector errors; uniform uses Lagrange's supports with equal weights.

5.3 Main Results

Table 2 reports the main comparison across the three flow-matching backbones. Every variant improves over its base model on all five metrics, and the best FlowBP variant exceeds every prior direct-gradient baseline on 14 of the 15 backbone–metric pairs, but the lead rotates rather than concentrating on any single variant. FlowBP-Lagrange takes four of five metrics on `SD3.5-M` and leads HPSv2.1 and UR-Align on `FLUX.1-dev`, while the reconstruction-based variants take PickScore and UR-IQ there and four of five on `FLUX.2-Klein-base`. Two metrics go elsewhere: UR-Align on `FLUX.2-Klein-base` (LeapAlign) and ImageReward on `FLUX.1-dev`, where the policy-gradient reference MixGRPO scores highest while losing UR-IQ relative to the base model. The preferred configuration thus depends on both the backbone and the evaluation criterion.

These gains are accompanied by improved overall compositional accuracy. On the GenEval benchmark (Ghosh et al., 2023), FlowBP-Lagrange lifts the `FLUX.1-dev` overall score from 63.25 to 69.88 and FlowBP-Bridge is second, with representative samples in Figure 2 and the task-level breakdown in Appendix B. With a stronger training reward, HPSv3++ (Liu et al., 2026), on `SD3.5-M`, the three FlowBP variants take the top three in-domain and held-out HPSv2.1 scores (Appendix B.6). Per-backbone evaluation dynamics over training are reported in Figure 7, and additional qualitative results across all three backbones in Appendix D.

5.4 Ablation Studies

The main-text ablations use `FLUX.1-dev`, which offers stable training dynamics at a representative model scale. We probe the four design axes of Section 4.2 through the concrete knobs of our variants: the reward-model input via endpoint reconstruction, the active set via the budget K, the integration weights via the quadrature rule, and the bridge coupling via the nested-gradient scale α.

Turning on one axis at a time. Table 3 adds one design choice per row. Endpoint reconstruction (row B) improves every metric with the active step and optimizer fixed, and produces the same pattern when applied to DRTune; the weight rule necessarily moves with it, from ReFL's σ_τ to Euler weights. A wider active set (row C) trades PickScore for gains on the other four metrics, while bridge coupling (row D) improves all five. Lagrange quadrature (row E) keeps the reward on the sampled $\boldsymbol{x}_0$ but replaces exact replay by a multi-support connector, reopening a potentially nonzero residual d_0; it gains on HPSv2.1 and UR-Align and gives back small amounts on the other three metrics.

Connector design. Figure 3 shows that the Lagrange connector has lower endpoint-prediction error than single-anchor Euler and equal-weight uniform connectors, together with higher and more stable reward. The support scale peaks at an intermediate value: either suppressing or overweighting non-anchor terms degrades training. Definitions are in Appendix A.7.

Nested-gradient scale and active-step budget. Figures 5 and 6 in Appendix B compare both knobs. Increasing K gives relatively consistent gains at the cost of longer training, with limited improvement beyond $K = 3$. Tuned bridge coupling can add a small gain, but is less reliable across settings. Its deployed scale ranges from $\alpha = 0.1$ on `FLUX.1-dev` to 1.0 on `FLUX.2-Klein-base` (Table 6).

Table 3: Cumulative effect of the four design axes on `FLUX.1-dev`. The starting point (A) is ReFL, and each row changes one design choice while keeping the preceding ones. Δ is the change over the row above. The last block transplants the endpoint axis onto a second baseline.

Configuration	In-Domain		Out-of-Domain							
	HPSv2.1		PickScore		ImageReward		UR-Align		UR-IQ	
	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)
Turning on one design axis at a time										
`FLUX.1-dev` (no post-training)	0.3016	—	22.5940	—	1.0094	—	3.3168	—	3.7551	—
A ReFL	0.3930	—	23.6886	—	1.3916	—	3.5005	—	4.0293	—
B + endpoint reconstruction	0.3993	+0.0063	**23.8121**	+0.1235	1.4117	+0.0201	3.5272	+0.0267	4.0532	+0.0239
C + multi-step active set (FlowBP-Sparse)	0.4081	+0.0088	23.7458	−0.0663	1.4742	+0.0625	3.5408	+0.0136	4.0897	+0.0365
D + bridge coupling (FlowBP-Bridge)	0.4119	+0.0038	23.7952	+0.0494	**1.5156**	+0.0414	3.5681	+0.0273	**4.1098**	+0.0201
E + high-order quadrature (FlowBP-Lagrange)	**0.4134**	+0.0015	23.7796	−0.0156	1.4906	−0.0250	**3.5848**	+0.0167	4.0950	−0.0148
Transplanting the endpoint axis onto another baseline										
DRTune	0.3920	—	23.6704	—	1.4273	—	3.5403	—	4.0697	—
+ endpoint reconstruction	0.4004	+0.0085	23.7039	+0.0335	1.4576	+0.0303	3.5645	+0.0242	4.0816	+0.0119

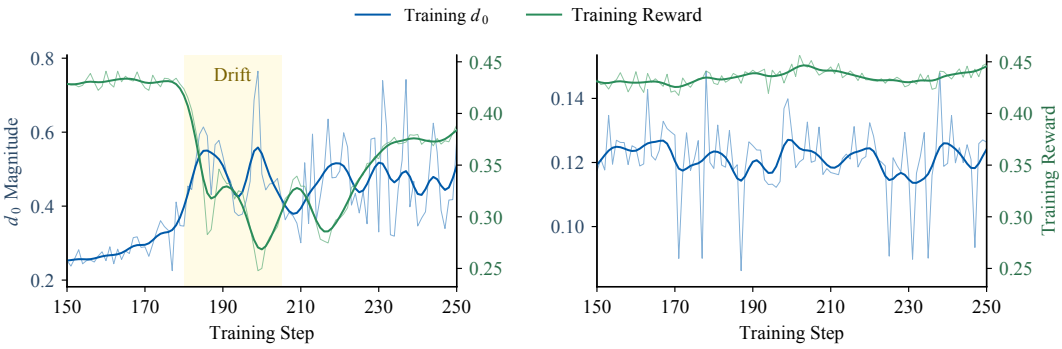


Figure 4: Connector residual and training reward. Left: LeapAlign's residual spike coincides with reward collapse (shaded). Right: FlowBP-Lagrange maintains smaller residuals without the same collapse.

Connector deviation. Figure 4 shows that spikes in LeapAlign's endpoint residual d_0 coincide with reward collapse, whereas the multi-support connector keeps d_0 smaller and shows no such collapse.

Recommended recipe. For routine use, we recommend **FlowBP-Sparse with** $K = 3$: evaluate the reward on the sampled endpoint, reconstruct it with Euler weights, and sample three active steps uniformly without bridge coupling. In the comparison runs ($K = 4$), FlowBP-Sparse exceeds every prior direct-gradient baseline on 12 of 15 backbone–metric pairs. Endpoint agreement alone is insufficient, as DRaFT-LV's direct target illustrates. Bridge coupling and Lagrange quadrature remain optional refinements, with the leading variant depending on backbone and metric (Table 2).

6 Conclusion

We introduced FlowBP, a surrogate-trajectory framework that separates the reward-model input, active support, integration weights, and bridge coupling. Its three instantiations evaluate rewards on sampled endpoints with only K active velocity calls and at most one nested state Jacobian. Our analysis establishes forward-error and worst-case amplification bounds in terms of effective coverage, connector residuals, and nested depth. Across three backbones, the best variant exceeds every prior direct-gradient baseline on 14 of 15 backbone–metric pairs.

Limitations. Our analysis focuses on Euler-based surrogate backward graphs, with experiments at 512×512 resolution. The stronger-reward and GenEval studies evaluate `SD3.5-M` and `FLUX.1-dev`, respectively. Broader evaluation across resolutions, reward models, and samplers remains future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with polishing the writing (grammar, phrasing, and LaTeX formatting) and with routine coding support such as plotting and refactoring of experiment scripts. We did not use generative AI tools for research ideation, deriving the theoretical results, or generating experimental results; the remaining categories requiring disclosure are not applicable to this work. We have reviewed all AI-assisted work: every AI-suggested edit to the text was checked by the authors against the intended technical meaning, and all AI-assisted code was read, executed, and verified against the reported numbers by at least two authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YCWjhGrJFD.

Black Forest Labs. FLUX.2 [klein]: Towards interactive visual intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence, 2026. Model weights: https://huggingface.co/black-forest-labs/FLUX.2-klein-base-9B.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1vmSEVL19f.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

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, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B.

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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=8OTPepXzeh.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Wbr51vK331.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

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. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.01569.

Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, and Liefeng Bo. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE. *arXiv preprint arXiv:2507.21802*, 2025.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13199–13208, June 2025.

Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. LeapAlign: Post-training flow matching models at any generation step by building two-step trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23238–23248, June 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Yijun Liu, Jie Huang, Zeyue Xue, Yuming Li, Ruizhe He, Haoran Li, Shijia Ge, and Siming Fu. HPSv3++: Scaling reward models across the full spectrum of diffusion model capabilities, 2026. URL https://arxiv.org/abs/2606.14657.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2023.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Xiangwei Shen, Zhimin Li, Zhantao Yang, Shiyi Zhang, Yingfang Zhang, Donghao Li, Chunyu Wang, Qinglin Lu, and Yansong Tang. Directly aligning the full diffusion trajectory with fine-grained human preference, 2025. URL https://arxiv.org/abs/2509.06942.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

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 (CVPR)*, pp. 8228–8238, June 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.

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.

Xiaoshi Wu, Yiming Hao, Manyuan Zhang, Keqiang Sun, Zhaoyang Huang, Guanglu Song, Yu Liu, and Hongsheng Li. Deep reward supervisions for tuning text-to-image diffusion models. In *European Conference on Computer Vision*, pp. 108–124. Springer, 2024.

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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=nLY2pOYBrJ.

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, 2024a.

Shentao Yang, Tianqi Chen, and Mingyuan Zhou. A dense reward view on aligning text-to-image diffusion with preference. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=xVXnXk9I3I.

Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=q3XavKPorV.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=YB9VGCClv9.

Xinchen Zhang, Ling Yang, Guohao Li, Yaqi Cai, Jiake Xie, Yong Tang, Yujiu Yang, Mengdi Wang, and Bin Cui. IterComp: Iterative composition-aware feedback learning from model gallery for text-to-image generation. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=4w99NAikOE.

Kaiwen Zheng, Yongxin Chen, Huayu Chen, Guande He, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Direct discriminative optimization: Your likelihood-based visual generative model is secretly a GAN discriminator. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=OJ6WE7F8tK.

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. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VJZ477R89F.

Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. DSPO: Direct score preference optimization for diffusion model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xyfb9HHvMe.

A Theoretical Analysis and Derivations

This appendix makes the algebraic claims of Section 4 precise. Appendices A.1 to A.3 define the computation graphs, the full-backpropagation reference, and three descriptive diagnostics. Appendix A.4 establishes the segment-error bounds, after which Appendices A.5 to A.8 derive the proposed updates and map prior methods into the same design space.

A.1 Notation, Conventions, and Standing Assumptions

Cached rollout. We follow the notation of Section 4. A no-gradient rollout of Equation (2) produces detached states and velocities $\{\boldsymbol{x}_i\}_{i=0}^{N}$ and $\{(\boldsymbol{v}_i, \sigma_i)\}_{i=1}^{N}$ with

$$\boldsymbol{x}_{i-1} = \boldsymbol{x}_i - h_i \boldsymbol{v}_i, \qquad \boldsymbol{v}_i := \boldsymbol{v}_\theta(\boldsymbol{x}_i, \sigma_i, \boldsymbol{c}), \qquad h_i := \sigma_i - \sigma_{i-1} > 0, \tag{10}$$

and $1 = \sigma_N > \sigma_{N-1} > \cdots > \sigma_0 = 0$, so that the step sizes telescope,

$$\sum_{i=1}^{N} h_i = \sigma_N - \sigma_0 = 1. \tag{11}$$

The surrogate replaces each cached velocity with $\tilde{\boldsymbol{v}}_i$ from Equation (4), parameterized by an active set $\mathcal{A} = \mathcal{A}_{\text{pre}} \cup \mathcal{A}_{\text{post}}$, a bridge index j with $0 \le j < k \le N$, and a nested-gradient scale $\alpha \in [0, 1]$. *All cached quantities are constants of the second, gradient-enabled forward pass*; in particular $\boldsymbol{x}_N$ is drawn independently of θ, so $\mathrm{d}\boldsymbol{x}_N/\mathrm{d}\theta = 0$, and the same holds for every $\boldsymbol{x}_i$ that a surrogate reads directly rather than reconstructs. Table 4 collects the symbols used below.

Table 4: Notation used in Appendix A. The first block is fixed by the rollout, the second by the surrogate, and the third by the comparison against full backpropagation.

Symbol	Meaning
$D,\ P$	dimensions of the latent $\boldsymbol{x} \in \mathbb{R}^D$ and of the parameters $\theta \in \mathbb{R}^P$
$N,\ \sigma_i,\ h_i$	rollout length, noise levels $1 = \sigma_N > \cdots > \sigma_0 = 0$, Euler steps $h_i = \sigma_i - \sigma_{i-1} > 0$
Δ	rollout grid width, $\Delta := \max_{1 \le i \le N} h_i$
$\boldsymbol{x}_i,\ \boldsymbol{v}_i$	cached state and cached velocity $\boldsymbol{v}_\theta(\boldsymbol{x}_i, \sigma_i, \boldsymbol{c})$ at index i
$k,\ j$	noisy-side anchor and bridge (split) index, $N \ge k > j \ge 0$
$\mathcal{A} = \mathcal{A}_{\text{pre}} \cup \mathcal{A}_{\text{post}}$	active set (velocities re-forwarded with gradients), $K := \lvert\mathcal{A}\rvert$
$\mathcal{S}_{\text{pre}}, \mathcal{S}_{\text{post}}$	quadrature supports of the two segments, $\mathcal{A}_{\text{pre}} \subseteq \mathcal{S}_{\text{pre}}$, $\mathcal{A}_{\text{post}} \subseteq \mathcal{S}_{\text{post}}$
M	number of supports *per segment*, $M = \lvert\mathcal{S}_{\text{pre}}\rvert = \lvert\mathcal{S}_{\text{post}}\rvert$
$w_i,\ w_i^{\text{L}}$	integration weight of support i; Euler weights are $w_i = h_i$, Lagrange weights w_i^{L} are given by Equation (24)
α	nested-gradient scale at the bridge
$\rho_i,\ \beta_i$	backward scale and effective backward weight of support i, with $\beta_i := \rho_i w_i$
$\tilde{\boldsymbol{v}}_i$	surrogate velocity of Equation (4)
$\hat{\boldsymbol{x}}_t,\ d_t$	surrogate estimate of $\boldsymbol{x}_t$ and its residual $d_t = \lVert\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\rVert_2$
$\partial_\theta \boldsymbol{v}_i,\ \partial_{\boldsymbol{x}} \boldsymbol{v}_i$	explicit Jacobians of $\boldsymbol{v}_\theta$ at the cached state, Equation (12)
$\mathbf{A}_l$	one-step Jacobian $\mathbf{I} - h_l \partial_{\boldsymbol{x}} \boldsymbol{v}_l \in \mathbb{R}^{D \times D}$ of the Euler map
$W_{\mathcal{A}}$	active-set coverage, Definition 5
d_{nest}	nested depth of the surrogate backward graph, Definition 6

Differentiation conventions. For a differentiable map $F : \mathbb{R}^P \to \mathbb{R}^D$ we write $\partial F/\partial\theta \in \mathbb{R}^{D \times P}$ for its Jacobian. Two derivatives of the velocity network must be kept apart, because the whole design space lives in the difference between them. We reserve $\mathrm{d}/\mathrm{d}\theta$ for *total* derivatives, which account for the dependence of a network input on θ, and ∂_θ and $\partial_{\boldsymbol{x}}$ for the *explicit* Jacobians of $\boldsymbol{v}_\theta$ at a frozen input:

$$\partial_\theta \boldsymbol{v}_i := \left.\frac{\partial\, \boldsymbol{v}_\theta(\boldsymbol{x}, \sigma_i, \boldsymbol{c})}{\partial\theta}\right|_{\boldsymbol{x}=\boldsymbol{x}_i} \in \mathbb{R}^{D \times P}, \qquad \partial_{\boldsymbol{x}} \boldsymbol{v}_i := \left.\frac{\partial\, \boldsymbol{v}_\theta(\boldsymbol{x}, \sigma_i, \boldsymbol{c})}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_i} \in \mathbb{R}^{D \times D}, \tag{12}$$

both evaluated at the cached state $\boldsymbol{x}_i$ and the current parameters. The quantity $\partial_\theta \boldsymbol{v}_i$ is exactly what reverse-mode autodiff returns for an active step whose input has been detached, which is why it is the only network derivative that appears in the direct terms of Equation (8). All products below are ordinary matrix products and do not commute.

Definition 1 (Stop-gradient)**.** sg is the identity on forward values and has zero derivative: for any tensor $\boldsymbol{y}$ in the graph, $\mathrm{sg}(\boldsymbol{y})$ carries the value of $\boldsymbol{y}$ and $\partial\,\mathrm{sg}(\boldsymbol{y})/\partial(\cdot) = 0$. It is not the derivative of any map on $\mathbb{R}^D$, so identities involving sg are statements about a computation graph, not about functions. Two consequences are used repeatedly:

(i) *(Connector, Equation (3).)* $\boldsymbol{x}_t \leftarrow \hat{\boldsymbol{x}}_t + \mathrm{sg}(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t)$ has forward value $\boldsymbol{x}_t$ and total derivative $\mathrm{d}\hat{\boldsymbol{x}}_t/\mathrm{d}\theta$.

(ii) *(Backward rescaling.)* $\mathrm{sg}(\boldsymbol{y}) + \rho\big(\boldsymbol{y} - \mathrm{sg}(\boldsymbol{y})\big)$ has forward value $\boldsymbol{y}$ and total derivative $\rho\,\mathrm{d}\boldsymbol{y}/\mathrm{d}\theta$ for any constant ρ.

Following LeapAlign, we reuse the symbol $\boldsymbol{x}_t$ for the post-connector latent; the occurrence inside $\mathrm{sg}(\cdot)$ always refers to the cached value.

Assumption (Rollout regularity). For the error analysis only, suppose the discrete cached velocities admit a vector-valued extension $\bar{\boldsymbol{v}} : [0,1] \to \mathbb{R}^D$ satisfying $\bar{\boldsymbol{v}}(\sigma_i) = \boldsymbol{v}_i$. We assume that

(A1) $\bar{\boldsymbol{v}}$ is L-Lipschitz in σ on $[0,1]$; in particular $\|\boldsymbol{v}_i - \boldsymbol{v}_l\|_2 \le L\,|\sigma_i - \sigma_l|$ for all cached indices;

(A2) $\bar{\boldsymbol{v}}$ is M times continuously differentiable in σ on each leap interval, with $\sup_\sigma \|\partial_\sigma^M \bar{\boldsymbol{v}}(\sigma)\|_2 \le L_M$;

(A3) $\|\partial_{\boldsymbol{x}} \boldsymbol{v}_i\|_2 \le G$ and $\|\partial_\theta \boldsymbol{v}_i\|_2 \le B$ for every i, where $\|\cdot\|_2$ is the spectral norm.

(A1) and (A2) are assumptions on a smooth extension of *one cached velocity sequence*, not on the network over all of $\mathbb{R}^D$. Stating the extension explicitly avoids identifying the discrete Euler states with an exact continuous ODE trajectory. (A3) is the bounded-Jacobian condition used for the amplification bound. (A1) is used in Appendix A.4, (A2) only in Lemma 10, and (A3) only in Proposition 3.

A.2 Full Backpropagation as the Reference Gradient

Full backpropagation through the Euler chain is the reference gradient that every surrogate replaces. We first state it exactly.

Proposition 2 (Exact gradient of the Euler rollout)**.** *Let $\boldsymbol{x}_0$ be produced by the N steps of Equation (2) with every step differentiated and with $\boldsymbol{x}_N$ drawn independently of θ. Then*

$$\left.\frac{\mathrm{d}\boldsymbol{x}_0}{\mathrm{d}\theta}\right|_{\mathrm{full}} \;=\; -\sum_{i=1}^{N} h_i\, J_i\, \partial_\theta \boldsymbol{v}_i, \qquad J_i := \mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_{i-1}, \qquad \mathbf{A}_l := \mathbf{I} - h_l\,\partial_{\boldsymbol{x}}\boldsymbol{v}_l, \tag{13}$$

where the factors of J_i are ordered left to right by increasing index and $J_1 = \mathbf{I}$ is the empty product.

Proof. Write $D_i := \mathrm{d}\boldsymbol{x}_i/\mathrm{d}\theta \in \mathbb{R}^{D\times P}$. Differentiating $\boldsymbol{x}_{i-1} = \boldsymbol{x}_i - h_i \boldsymbol{v}_\theta(\boldsymbol{x}_i, \sigma_i, \boldsymbol{c})$ gives

$$\begin{aligned} D_{i-1} &= D_i - h_i \frac{\mathrm{d}}{\mathrm{d}\theta}\boldsymbol{v}_\theta(\boldsymbol{x}_i, \sigma_i, \boldsymbol{c}) = D_i - h_i\big(\partial_\theta \boldsymbol{v}_i + \partial_{\boldsymbol{x}}\boldsymbol{v}_i D_i\big) \\ &= \mathbf{A}_i\, D_i - h_i\,\partial_\theta \boldsymbol{v}_i, \end{aligned}$$

where the middle equality is the chain rule. Starting from $D_N = 0$ and recursively substituting this identity yields, for $0 \le m < N$,

$$D_m = -\sum_{i=m+1}^{N} h_i \Big(\textstyle\prod_{l=m+1}^{i-1} \mathbf{A}_l\Big)\partial_\theta \boldsymbol{v}_i.$$

The product is the identity when its index set is empty. Setting $m = 0$ gives Equation (13). □

From latent Jacobian to reward gradient. What a method actually applies to θ is the reward gradient, obtained from Proposition 2 by one more chain rule,

$$\nabla_\theta\, r(\boldsymbol{x}_0, \boldsymbol{c}) \;=\; \left(\left.\frac{\mathrm{d}\boldsymbol{x}_0}{\mathrm{d}\theta}\right|_{\mathrm{full}}\right)^{\top} \nabla_{\boldsymbol{x}} r(\boldsymbol{x}_0, \boldsymbol{c}) \;\in\; \mathbb{R}^P. \tag{14}$$

A surrogate may change either factor of Equation (14). Let $\boldsymbol{y}_0$ denote the tensor value actually read by the reward model, and let D_{surr} denote the endpoint Jacobian delivered by the surrogate backward graph. Its reward gradient is

$$\nabla_\theta^{\text{surr}} r \;=\; D_{\text{surr}}^\top \nabla_{\boldsymbol{x}} r(\boldsymbol{y}_0, \boldsymbol{c}). \tag{15}$$

The forward input and backward endpoint Jacobian are distinct choices:

	$\boldsymbol{y}_0$	D_{surr}
detached predictor	$\hat{\boldsymbol{x}}_0$	$\partial\hat{\boldsymbol{x}}_0/\partial\theta$
endpoint connector	$\boldsymbol{x}_0$	$\partial\hat{\boldsymbol{x}}_0/\partial\theta$
Euler reconstruction	$\boldsymbol{x}_0$	$\partial\boldsymbol{x}_0^{\text{recon}}/\partial\theta$

where $\boldsymbol{x}_0^{\text{recon}}$ has the cached endpoint as its forward value. The detached similarity weight of Appendix B only multiplies Equation (15) by a sample-dependent scalar and is omitted below.

Relative to Equation (13), a surrogate changes the effective support $\{\beta_i\}$, the forward/backward endpoint pair, and the depth of the retained state-Jacobian product. Appendix A.3 defines one diagnostic for each property; none is a distance to the full-backpropagation gradient.

Truncating J_i is not only a loss of information; it is also what keeps the surrogate gradient bounded. The following makes quantitative the informal statement in Section 4.2 that a single Jacobian factor caps amplification.

Proposition 3 (Worst-case gradient amplification)**.** *Under (A3), let $\beta_i = \rho_i w_i$ denote the possibly signed effective backward weights and set $W_{\mathcal{A}} = \sum_{i\in\mathcal{A}} |\beta_i|$. Then:*

(i) $\|J_i\|_2 \le \prod_{l<i}\big(1 + h_l G\big) \le e^{G\sigma_{i-1}}$*, and therefore* $\big\|\mathrm{d}\boldsymbol{x}_0/\mathrm{d}\theta|_{\text{full}}\big\|_2 \le B\, e^{G\sigma_{N-1}}$*;*

(ii) the surrogate gradient of Equation (8) satisfies $\big\|\partial\boldsymbol{x}_0/\partial\theta|_{\text{surr}}\big\|_2 \le B\big(1 + \alpha|\beta_j|G\big)W_{\mathcal{A}}$*, with $\beta_j := 0$ when there is no bridge.*

Proof. For (i), submultiplicativity of the spectral norm and (A3) give $\|J_i\|_2 \le \prod_{l<i} \|\mathbf{I} - h_l \partial_{\boldsymbol{x}} \boldsymbol{v}_l\|_2 \le \prod_{l<i}(1 + h_l G)$, and $1 + u \le e^u$ together with $\sum_{l<i} h_l = \sigma_{i-1}$ yields $\|J_i\|_2 \le e^{G\sigma_{i-1}}$. Substituting into Equation (13) and using Equation (11),

$$\Big\| \tfrac{\mathrm{d}\boldsymbol{x}_0}{\mathrm{d}\theta}\big|_{\text{full}} \Big\|_2 \le \sum_{i=1}^{N} h_i \, \|J_i\|_2 \, \|\partial_\theta \boldsymbol{v}_i\|_2 \le B\, e^{G\sigma_{N-1}} \sum_{i=1}^{N} h_i = B\, e^{G\sigma_{N-1}}.$$

For (ii), the triangle inequality applied to Equation (8) gives

$$\Big\| \tfrac{\partial\boldsymbol{x}_0}{\partial\theta}\big|_{\text{surr}} \Big\|_2 \le B \sum_{i\in\mathcal{A}} |\beta_i| + \alpha|\beta_j| G B \sum_{i\in\mathcal{A}_{\text{pre}}} |\beta_i| \le B\big(1 + \alpha|\beta_j|G\big)W_{\mathcal{A}}. \qquad \square$$

The proposition compares worst-case upper bounds, not the realized gradient norm. Over the fixed interval $[0, 1]$, full backpropagation has an exponential bound in the accumulated state-Jacobian magnitude, whereas the surrogate bound is affine in the magnitude of its single retained state Jacobian. It does not claim exponential growth in the number of Euler steps N. Setting $\alpha = 0$ removes the G-dependent term from the surrogate bound.

Remark 4 (Activation memory). The cached pass stores $O(N)$ latents but no network activations; only the K re-forwarded calls retain activations, versus $O(N)$ calls for full backpropagation.

A.3 Three Descriptive Diagnostics

The following diagnostics summarize complementary properties of a surrogate and are not combined into a gradient-fidelity score.

Definition 5 (Coverage)**.** The effective coverage of an active set is the total absolute backward weight carried by its steps,

$$W_{\mathcal{A}} = \sum_{i\in\mathcal{A}} |\beta_i|, \qquad \beta_i := \rho_i w_i, \qquad \text{normalized by} \quad \sum_{i=1}^{N} h_i = \sigma_N = 1, \tag{16}$$

where ρ_i includes every stop-gradient rescaling applied to support i. The absolute value makes the definition meaningful for signed quadrature weights and is the quantity entering Proposition 3. Full backpropagation has $\beta_i = h_i > 0$ for all steps and hence $W_{\mathcal{A}} = 1$.

Coverage records effective weight mass only. It does not record the locations of active supports, the values of $\partial_\theta \boldsymbol{v}_i$, or the state-Jacobian products J_i; consequently $W_{\mathcal{A}} = 1$ need not reproduce full backpropagation. The implementation-level scales entering ρ_i are specified in Remark 17.

The residual $d_t = \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2$ is likewise a forward diagnostic. For a posterior predictor it is the displacement of the reward input from the cached sample; for a connector it is the discrepancy between the cached forward value and the surrogate value whose derivative is used. It is zero for exact Euler reconstruction. Without additional C^1 assumptions, d_t alone does not bound a Jacobian or reward-gradient error.

Definition 6 (Nested depth)**.** Consider the surrogate backward graph, the directed acyclic graph whose nodes are the tensors created by the gradient-enabled forward pass, with an edge $u \to v$ whenever the local Jacobian $\partial v/\partial u$ is nonzero; edges through sg and zero-scaled edges are therefore absent. Define d_{nest} as the maximum, over directed paths in this graph from θ to the reward input, of the number of edges of the form $\boldsymbol{x} \mapsto \boldsymbol{v}_\theta(\boldsymbol{x}, \cdot, \boldsymbol{c})$, i.e. the number of $\partial_{\boldsymbol{x}} \boldsymbol{v}$ factors in the path's Jacobian product.

Nested depth records the structural length of state-Jacobian products. Full backpropagation has $d_{\mathrm{nest}} = N - 1$: the path that enters through $\boldsymbol{v}_N$ traverses the edges $\boldsymbol{x}_{N-1} \mapsto \boldsymbol{v}_{N-1}, \ldots, \boldsymbol{x}_1 \mapsto \boldsymbol{v}_1$, matching the $N - 1$ factors of J_N in Equation (13). Detached surrogates have $d_{\mathrm{nest}} = 0$, and bridged surrogates with $\alpha \neq 0$ have $d_{\mathrm{nest}} = 1$. This count does not assess the numerical value of the retained Jacobian or the quality of its resulting gradient.

A.4 Forward Error of a Surrogate Segment

The residuals d_0 and d_j of Appendix A.3 are quadrature errors of the surrogate against the *cached rollout*, and their orders are what Table 1 reports in its "forward error" column. Fix two indices $0 \le t < s \le N$ and write

$$I := (t, s] = \{t+1, \ldots, s\}, \qquad h := \sigma_s - \sigma_t = \sum_{i \in I} h_i, \qquad \Delta = \max_{1 \le i \le N} h_i. \tag{17}$$

Telescoping Equation (10) over I gives the reference $\boldsymbol{x}_t = \boldsymbol{x}_s - \sum_{i \in I} h_i \boldsymbol{v}_i$, whereas a surrogate rule with support $\mathcal{S} \subseteq I$ and weights $\{w_i\}_{i \in \mathcal{S}}$ produces

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_s - \sum_{i \in \mathcal{S}} w_i \, \tilde{\boldsymbol{v}}_i, \qquad d_t := \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2. \tag{18}$$

The first lemma states the property that makes d_t a function of the rule alone, and that makes Euler reconstruction connector-free.

Lemma 7 (Forward consistency of the surrogate velocities)**.** *Assume deterministic re-evaluation with unchanged parameters and model settings. In exact arithmetic, every $\tilde{\boldsymbol{v}}_i$ of Equation (4) has forward value $\boldsymbol{v}_i$, provided the latent feeding the bridge case has forward value $\boldsymbol{x}_j$. Consequently the forward value of Equation (18) is $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_s - \sum_{i \in \mathcal{S}} w_i \boldsymbol{v}_i$, so d_t depends only on the rule $(\mathcal{S}, \{w_i\})$ and the cached velocities, and not on which supports are active.*

Proof. Inactive supports equal $\mathrm{sg}(\boldsymbol{v}_i)$; regular active supports re-evaluate the same network at $\mathrm{sg}(\boldsymbol{x}_i)$; and the bridge input has forward value $\alpha \boldsymbol{x}_j + (1 - \alpha)\boldsymbol{x}_j = \boldsymbol{x}_j$. Thus every case has value $\boldsymbol{v}_i$. □

Corollary 8 (Exact Euler replay)**.** *If $\mathcal{S} = I$ and $w_i = h_i$ for all $i \in I$, then $d_t = 0$.*

Proof. By Lemma 7, $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_s - \sum_{i \in I} h_i \boldsymbol{v}_i = \boldsymbol{x}_t$. □

This covers FlowBP-Sparse on $(N \to 0)$ and FlowBP-Bridge on each of $(N \to j)$ and $(j \to 0)$; it also covers DRaFT-LV, for which $I = \{1\} = \mathcal{S}$ and $w_1 = \sigma_1 - \sigma_0 = h_1$, so its single leap coincides with the last Euler step.

Lemma 9 (One-point leap)**.** *If $\mathcal{S} = \{s\}$ and $w_s = h$, then under (A1)*

$$\begin{aligned} d_t &= \Big\| \sum_{i \in I} h_i \, (\boldsymbol{v}_i - \boldsymbol{v}_s) \Big\|_2 \;\le\; L \sum_{i \in I} h_i \, (\sigma_s - \sigma_i) \\ &= \frac{L}{2} \Big(h^2 - \sum_{i \in I} h_i^2 \Big) \;\le\; \frac{L}{2} \Big(1 - \frac{1}{|I|} \Big) h^2 \;=\; O(h^2). \end{aligned} \tag{19}$$

Proof. Since $\sum_{i\in I} h_i = h$, Lemma 7 gives $\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t = -\sum_{i\in I} h_i \boldsymbol{v}_i + h\boldsymbol{v}_s = \sum_{i\in I} h_i(\boldsymbol{v}_s - \boldsymbol{v}_i)$, which proves the equality. Assumption (A1) and the triangle inequality give the first bound. The identity follows by collecting pairs, $\sum_i h_i(\sigma_s - \sigma_i) = \sum_{i<l} h_i h_l = \frac{1}{2}(h^2 - \sum_i h_i^2)$, and Cauchy–Schwarz gives $\sum_i h_i^2 \geq h^2/|I|$. □

So a single-velocity leap costs $O(h^2)$, with an explicit constant that grows quadratically in the interval it spans. LeapAlign pays this on both of its leaps; ReFL and DRTune pay it once, in the case $t = 0$ (hence $h = \sigma_s$ and $w_s = \sigma_s$) with $s = \tau$ and $s = t_{\min}$ respectively.

Lemma 10 (Interpolatory rule of order M)**.** *Let $\mathcal{S} \subseteq I$ with $|\mathcal{S}| = M$, and let $\{w_i\}_{i\in\mathcal{S}}$ be the integrated Lagrange weights of Equation (24). Under (A1) and (A2),*

$$d_t \;\leq\; \underbrace{\frac{L_M}{M!}\,h^{M+1}}_{\text{quadrature error}} \;+\; \underbrace{\frac{L}{2}\,h\Delta}_{\text{Riemann-sum error}} \;=\; O\big(h^{M+1} + h\Delta\big)\,. \tag{20}$$

Proof. Insert the comparison integral $\int_{\sigma_t}^{\sigma_s} \bar{\boldsymbol{v}}(\sigma)\,\mathrm{d}\sigma$ and apply the triangle inequality:

$$\begin{aligned} d_t &= \Big\|\sum_{i\in I} h_i \boldsymbol{v}_i - \sum_{i\in\mathcal{S}} w_i \boldsymbol{v}_i\Big\|_2 \\ &\leq \underbrace{\Big\|\int_{\sigma_t}^{\sigma_s} \bar{\boldsymbol{v}}(\sigma)\,\mathrm{d}\sigma - \sum_{i\in\mathcal{S}} w_i \boldsymbol{v}_i\Big\|_2}_{(a)} + \underbrace{\Big\|\sum_{i\in I} h_i \boldsymbol{v}_i - \int_{\sigma_t}^{\sigma_s} \bar{\boldsymbol{v}}(\sigma)\,\mathrm{d}\sigma\Big\|_2}_{(b)}. \end{aligned}$$

Term (a) is the error of integrating the degree-$(M-1)$ Lagrange interpolant of $\bar{\boldsymbol{v}}$ through the nodes $\{\sigma_i\}_{i\in\mathcal{S}}$. For a vector-valued function, the Hermite–Genocchi representation of the divided difference gives

$$\|\bar{\boldsymbol{v}}(\sigma) - P_{M-1}(\sigma)\|_2 \leq \frac{L_M}{M!}\prod_{q\in\mathcal{S}} |\sigma - \sigma_q| \leq \frac{L_M}{M!} h^M.$$

Integrating this pointwise bound over an interval of length h yields $(a) \leq L_M h^{M+1}/M!$. Term (b) is the error of the right-endpoint rectangle rule applied to the chosen extension: by (A1),

$$(b) = \Big\|\sum_{i\in I}\int_{\sigma_{i-1}}^{\sigma_i} \big(\boldsymbol{v}_i - \bar{\boldsymbol{v}}(\sigma)\big)\mathrm{d}\sigma\Big\|_2 \leq \sum_{i\in I}\int_{\sigma_{i-1}}^{\sigma_i} L\,(\sigma_i - \sigma)\,\mathrm{d}\sigma = \frac{L}{2}\sum_{i\in I} h_i^2 \leq \frac{L}{2}\,\Delta\,h. \qquad \square$$

Remark 11 (A grid-dependent term in the bound). The $O(h\Delta)$ term arises because the proof compares the discrete Euler sum with an auxiliary integral; it is an upper-bound term, not an error floor. Moreover, L_M may vary with M, so the bound does not imply monotone improvement with interpolation order.

A.5 FlowBP-Sparse

FlowBP-Sparse drops the bridge ($j = 0$, $\alpha = 0$, $\mathcal{A}_{\text{post}} = \emptyset$) and reconstructs the endpoint by replaying the cached trajectory with Euler weights. With no bridge the bridge case of Equation (4) does not arise: every $i \in \mathcal{A} = \mathcal{A}_{\text{pre}}$ uses the regular active form $\boldsymbol{v}_\theta(\mathrm{sg}(\boldsymbol{x}_i), \sigma_i, \boldsymbol{c})$ and every $i \notin \mathcal{A}$ uses $\mathrm{sg}(\boldsymbol{v}_i)$.

Proposition 12 (Gradient of FlowBP-Sparse)**.** *Consider first the unscaled graph, $\rho_i = 1$ for active supports. Let*

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_N - \sum_{i=1}^{N} h_i\,\tilde{\boldsymbol{v}}_i.$$

Then $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0$ as a forward value, so no endpoint connector is needed, and

$$\frac{\partial \boldsymbol{x}_0}{\partial \theta} = -\sum_{i\in\mathcal{A}} h_i\,\partial_\theta \boldsymbol{v}_i. \tag{21}$$

Proof. The forward statement is Corollary 8 with $t = 0$, $s = N$ and $\mathcal{S} = I = \{1, \dots, N\}$. Since $\boldsymbol{x}_N$ is cached, inactive calls have zero derivative, and active inputs are detached,

$$\frac{\partial \boldsymbol{x}_0}{\partial \theta} = -\sum_{i=1}^{N} h_i \frac{\partial \tilde{\boldsymbol{v}}_i}{\partial \theta} = -\sum_{i \in \mathcal{A}} h_i \, \partial_\theta \boldsymbol{v}_i.$$

□

Inactive velocities complete the exact forward replay but contribute no gradient. All active inputs are detached, so $d_{\text{nest}} = 0$ and Proposition 3(ii) gives the bound $BW_{\mathcal{A}}$, independent of G.

A.6 FLOWBP-BRIDGE

FlowBP-Bridge adds a single bridge to FlowBP-Sparse's full Euler-quadrature trajectory. The split index $j \in \{1, \dots, N-1\}$ partitions the rollout into a pre-segment $(N \to j)$ and a post-segment $(j \to 0)$. The active sets are $\mathcal{A}_{\text{pre}} \subseteq \{j+1, \dots, N\}$ and $\mathcal{A}_{\text{post}} \subseteq \{1, \dots, j\}$ with the bridge anchor $j \in \mathcal{A}_{\text{post}}$, so that $\mathcal{A} = \mathcal{A}_{\text{pre}} \cup \mathcal{A}_{\text{post}}$ is a disjoint union. All weights are Euler weights $w_i = h_i$, and each $\tilde{\boldsymbol{v}}_i$ follows Equation (4): the regular active form for $i \in \mathcal{A}_{\text{pre}}$ and $i \in \mathcal{A}_{\text{post}} \setminus \{j\}$, and the bridge case at $i = j$.

Proposition 13 (Gradient of FlowBP-Bridge)**.** *Consider first the unscaled graph, $\rho_i = 1$ for active supports. Let the two segments be reconstructed as*

$$\boldsymbol{x}_j = \boldsymbol{x}_N - \sum_{i=j+1}^{N} h_i \, \tilde{\boldsymbol{v}}_i, \qquad \boldsymbol{x}_0 = \boldsymbol{x}_j - \sum_{i=1}^{j} h_i \, \tilde{\boldsymbol{v}}_i.$$

Then both reconstructions match the cached states in the forward pass, and

$$\frac{\partial \boldsymbol{x}_0}{\partial \theta} = -\sum_{i \in \mathcal{A}} h_i \, \partial_\theta \boldsymbol{v}_i \; + \; \alpha \, h_j \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j \sum_{i \in \mathcal{A}_{\text{pre}}} h_i \, \partial_\theta \boldsymbol{v}_i. \tag{22}$$

Proof. Exact replay on $(N \to j)$ and $(j \to 0)$ gives the cached forward values. Write $D_t := \partial \boldsymbol{x}_t / \partial \theta$. Since only active pre-segment calls carry gradients and their inputs are detached,

$$D_j = -\sum_{i \in \mathcal{A}_{\text{pre}}} h_i \, \partial_\theta \boldsymbol{v}_i.$$

At the bridge, the input $\alpha \boldsymbol{x}_j + (1-\alpha)\,\mathrm{sg}(\boldsymbol{x}_j)$ has derivative αD_j. Therefore

$$\begin{aligned} D_0 &= D_j - \sum_{i \in \mathcal{A}_{\text{post}} \setminus \{j\}} h_i \, \partial_\theta \boldsymbol{v}_i - h_j \, \partial_\theta \boldsymbol{v}_j - \alpha h_j \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j D_j \\ &= -\sum_{i \in \mathcal{A}_{\text{pre}}} h_i \, \partial_\theta \boldsymbol{v}_i - \sum_{i \in \mathcal{A}_{\text{post}}} h_i \, \partial_\theta \boldsymbol{v}_i + \alpha h_j \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j \sum_{i \in \mathcal{A}_{\text{pre}}} h_i \, \partial_\theta \boldsymbol{v}_i \\ &= -\sum_{i \in \mathcal{A}} h_i \, \partial_\theta \boldsymbol{v}_i + \alpha h_j \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j \sum_{i \in \mathcal{A}_{\text{pre}}} h_i \, \partial_\theta \boldsymbol{v}_i, \end{aligned}$$

where the last equality uses $\mathcal{A}_{\text{pre}} \cap \mathcal{A}_{\text{post}} = \emptyset$. □

Equation (22) is the case $\beta_i = w_i = h_i$ of Equation (8). Setting $\alpha = 0$ removes the nested term, while $\alpha = 1$ retains the full one-Jacobian bridge. Exact replay constructs the split latent, so $d_j = 0$ by Corollary 8.

A.7 FLOWBP-LAGRANGE

FlowBP-Lagrange keeps the two-segment connector topology of LeapAlign but replaces each one-point leap by an integrated polynomial approximation of the velocity field, i.e. it moves from Lemma 9 to Lemma 10. Let $k > j > 0$ be the sampled bridge indices. The quadrature supports are

$$\mathcal{S}_{\text{pre}} \subseteq \{j+1, \dots, k\} \;\text{ with } k \in \mathcal{S}_{\text{pre}}, \qquad \mathcal{S}_{\text{post}} \subseteq \{1, \dots, j\} \;\text{ with } j \in \mathcal{S}_{\text{post}}, \tag{23}$$

and we write $M = |\mathcal{S}_{\mathrm{pre}}| = |\mathcal{S}_{\mathrm{post}}|$ for the number of supports per segment. The gradient-active sets satisfy $\mathcal{A}_{\mathrm{pre}} \subseteq \mathcal{S}_{\mathrm{pre}}$ and $\mathcal{A}_{\mathrm{post}} \subseteq \mathcal{S}_{\mathrm{post}}$, and we require both anchors to be active, $k \in \mathcal{A}_{\mathrm{pre}}$ and $j \in \mathcal{A}_{\mathrm{post}}$, so that $K = |\mathcal{A}| \le 2M$. Inactive supports enter only the forward quadrature, through detached cached velocities. The Lagrange basis polynomial associated with a support $s_m \in \mathcal{S}_{\mathrm{pre}}$ and its integrated weight are

$$\ell_m^{\mathcal{S}_{\mathrm{pre}}}(\sigma) = \prod_{n \neq m} \frac{\sigma - \sigma_{s_n}}{\sigma_{s_m} - \sigma_{s_n}}, \qquad w_{s_m}^{\mathrm{L}} = \int_{\sigma_j}^{\sigma_k} \ell_m^{\mathcal{S}_{\mathrm{pre}}}(\sigma)\, \mathrm{d}\sigma, \tag{24}$$

and the post-segment quantities $\ell_m^{\mathcal{S}_{\mathrm{post}}}$ and $w_{s'_m}^{\mathrm{L}}$ are constructed identically over $[\sigma_0, \sigma_j]$. We drop the segment superscript on w_i^{L}, since $\mathcal{S}_{\mathrm{pre}}$ and $\mathcal{S}_{\mathrm{post}}$ are disjoint and the segment is determined by i.

Lemma 14 (Consistency and sign of the Lagrange weights)**.** *The weights of Equation (24) reproduce the length of their segment exactly,*

$$\sum_{i \in \mathcal{S}_{\mathrm{pre}}} w_i^{\mathrm{L}} = \sigma_k - \sigma_j, \qquad \sum_{i \in \mathcal{S}_{\mathrm{post}}} w_i^{\mathrm{L}} = \sigma_j - \sigma_0 = \sigma_j, \tag{25}$$

so the two segments together carry total weight σ_k. They are not non-negative in general.

Proof. The Lagrange basis is a partition of unity, $\sum_m \ell_m^{\mathcal{S}}(\sigma) \equiv 1$, because the interpolant of the constant function 1 through any set of distinct nodes is 1; integrating over the segment gives Equation (25). For the second claim it suffices to exhibit a configuration with a negative weight. Take $M = 2$ on a segment $[\sigma_t, \sigma_s]$ with nodes $\sigma_a < \sigma_s$, and write $p = \sigma_s - \sigma_a$ for the node span and $q = \sigma_a - \sigma_t$ for the extrapolated part. Direct integration gives

$$w_a = \frac{(p+q)^2}{2p} > 0, \qquad w_s = \frac{p^2 - q^2}{2p},$$

which sum to $p + q = h$ as required and satisfy $w_s < 0$ whenever $q > p$. □

The sign caveat is intrinsic to the construction: the supports lie in $(t, s]$ while the integral starts at σ_t, below the smallest node, so the rule may extrapolate on $[\sigma_t, \sigma_{\min \mathcal{S}}]$. We therefore make no positivity claim. The forward consistency identity Equation (25) uses signed weights, while effective coverage and the amplification bound use their absolute effective values $|\rho_i w_i^{\mathrm{L}}|$.

Effective backward weights. The surrogate velocity $\tilde{\boldsymbol{v}}_i$ at an active support is given by Equation (4): the post-segment anchor $i = j$ uses the bridge case $\boldsymbol{v}_\theta(\alpha \boldsymbol{x}_j + (1-\alpha)\,\mathrm{sg}(\boldsymbol{x}_j), \sigma_j, \boldsymbol{c})$, and every other active support uses the regular active case $\boldsymbol{v}_\theta(\mathrm{sg}(\boldsymbol{x}_i), \sigma_i, \boldsymbol{c})$. Define the backward scale ρ_i to be 1 at the two anchors, $g_s \in [0, 1]$ at non-anchor active supports, and 0 at inactive supports. Writing

$$\beta_i := \rho_i w_i^{\mathrm{L}}, \qquad \tilde{\boldsymbol{v}}_i^{\rho} = \mathrm{sg}(\tilde{\boldsymbol{v}}_i) + \rho_i \big(\tilde{\boldsymbol{v}}_i - \mathrm{sg}(\tilde{\boldsymbol{v}}_i)\big),$$

inactive supports reduce to $\tilde{\boldsymbol{v}}_i^{\rho} = \mathrm{sg}(\boldsymbol{v}_i)$, and by Definition 1 the forward quadrature is unchanged for any ρ_i, while the backward contribution uses the effective, possibly signed weight β_i.

Proposition 15 (Gradient of FlowBP-Lagrange)**.** *Let the two segments be connected to the cached trajectory by*

$$\hat{\boldsymbol{x}}_j = \boldsymbol{x}_k - \sum_{i \in \mathcal{S}_{\mathrm{pre}}} w_i^{\mathrm{L}}\, \tilde{\boldsymbol{v}}_i^{\rho}, \qquad \boldsymbol{x}_j = \hat{\boldsymbol{x}}_j + \mathrm{sg}(\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j),$$

$$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_j - \sum_{i \in \mathcal{S}_{\mathrm{post}}} w_i^{\mathrm{L}}\, \tilde{\boldsymbol{v}}_i^{\rho}, \qquad \boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0 + \mathrm{sg}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0).$$

The just-pinned $\boldsymbol{x}_j$ is the latent used by the bridge call $\tilde{\boldsymbol{v}}_j$ in Equation (4). Then

$$\frac{\partial \boldsymbol{x}_0}{\partial \theta} = -\sum_{i \in \mathcal{A}} \beta_i\, \partial_\theta \boldsymbol{v}_i \;+\; \alpha\, \beta_j\, \partial_{\boldsymbol{x}} \boldsymbol{v}_j \sum_{i \in \mathcal{A}_{\mathrm{pre}}} \beta_i\, \partial_\theta \boldsymbol{v}_i. \tag{26}$$

Proof. Write $D_t := \partial \boldsymbol{x}_t / \partial \theta$. The connectors preserve the surrogate derivatives, $\boldsymbol{x}_k$ is cached, and inactive supports have $\rho_i = 0$. Hence

$$D_j = -\sum_{i \in \mathcal{A}_{\text{pre}}} \beta_i \, \partial_\theta \boldsymbol{v}_i.$$

The pinned split has forward value $\boldsymbol{x}_j$, so the bridge Jacobians are evaluated at the cached state. Separating the bridge support then gives

$$\begin{aligned} D_0 &= D_j - \sum_{i \in \mathcal{A}_{\text{post}} \setminus \{j\}} \beta_i \, \partial_\theta \boldsymbol{v}_i - \beta_j \, \partial_\theta \boldsymbol{v}_j - \alpha \beta_j \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j D_j \\ &= -\sum_{i \in \mathcal{A}_{\text{pre}}} \beta_i \, \partial_\theta \boldsymbol{v}_i - \sum_{i \in \mathcal{A}_{\text{post}}} \beta_i \, \partial_\theta \boldsymbol{v}_i + \alpha \beta_j \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j \sum_{i \in \mathcal{A}_{\text{pre}}} \beta_i \, \partial_\theta \boldsymbol{v}_i \\ &= -\sum_{i \in \mathcal{A}} \beta_i \, \partial_\theta \boldsymbol{v}_i + \alpha \beta_j \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j \sum_{i \in \mathcal{A}_{\text{pre}}} \beta_i \, \partial_\theta \boldsymbol{v}_i, \end{aligned}$$

where the last equality uses $\mathcal{A}_{\text{pre}} \cap \mathcal{A}_{\text{post}} = \emptyset$. □

Equation (26) is the unified gradient Equation (8) with effective weights $\beta_i = \rho_i w_i^{\text{L}}$. The last term is the only nested one: it contains exactly one Jacobian factor and is scaled by α, so $d_{\text{nest}} = 1$ for $\alpha \beta_j \neq 0$, irrespective of M.

Corollary 16 (LeapAlign is the one-support case)**.** *If $\mathcal{S}_{\text{pre}} = \mathcal{A}_{\text{pre}} = \{k\}$ and $\mathcal{S}_{\text{post}} = \mathcal{A}_{\text{post}} = \{j\}$, then $w_k^{\text{L}} = \sigma_k - \sigma_j$, $w_j^{\text{L}} = \sigma_j$, all $\rho_i = 1$, and Equation (26) reduces to the LeapAlign gradient*

$$\frac{\partial \boldsymbol{x}_0}{\partial \theta} = -(\sigma_k - \sigma_j) \, \partial_\theta \boldsymbol{v}_k - \sigma_j \, \partial_\theta \boldsymbol{v}_j + \alpha \, \sigma_j \, (\sigma_k - \sigma_j) \, \partial_{\boldsymbol{x}} \boldsymbol{v}_j \, \partial_\theta \boldsymbol{v}_k. \tag{27}$$

Proof. With a single node the basis polynomial is the empty product $\ell_1 \equiv 1$, so $w_k^{\text{L}} = \int_{\sigma_j}^{\sigma_k} \mathrm{d}\sigma = \sigma_k - \sigma_j$ and $w_j^{\text{L}} = \int_{\sigma_0}^{\sigma_j} \mathrm{d}\sigma = \sigma_j$; both supports are anchors, so $\rho_k = \rho_j = 1$. Substituting into Equation (26) gives Equation (27). □

A.8 Mapping Prior Methods to the Design Axes

The notation above also describes the prior updates in Table 1; complete computation graphs are given in Appendix C.

ReFL takes $j = 0$ (no bridge), a single active step at a randomly sampled tail index τ, and the one-point rule of Lemma 9 with $t = 0$, $s = \tau$, hence $w_\tau = \sigma_\tau$. The reward reads the model-implied clean estimate $\hat{\boldsymbol{x}}_0^{(\tau)} = \boldsymbol{x}_\tau - \sigma_\tau \tilde{\boldsymbol{v}}_\tau$. For an unguided predictor at the population optimum of the rectified-flow regression objective this equals $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_\tau, \boldsymbol{c}]$; for a finite learned model it is an approximation. Its structural gradient is

$$\frac{\partial \hat{\boldsymbol{x}}_0^{(\tau)}}{\partial \theta} = -\sigma_\tau \, \partial_\theta \boldsymbol{v}_\tau, \qquad d_{\text{nest}} = 0, \qquad d_0 \leq \tfrac{L}{2} \sigma_\tau^2.$$

DRaFT-LV uses the exact final-step target $\boldsymbol{z}_0 = \boldsymbol{x}_1 - \sigma_1 \tilde{\boldsymbol{v}}_1$, where $\sigma_1 = h_1$, and additionally forms detached re-noised states $\boldsymbol{x}_1^{(q)}$ with targets $\boldsymbol{z}_q = \boldsymbol{x}_1^{(q)} - \sigma_1 \boldsymbol{v}_\theta(\mathrm{sg}(\boldsymbol{x}_1^{(q)}), \sigma_1, \boldsymbol{c})$ for $q = 1, \ldots, n$. Their Jacobians are

$$\begin{aligned} \frac{\partial \boldsymbol{z}_0}{\partial \theta} &= -h_1 \, \partial_\theta \boldsymbol{v}_1, \\ \frac{\partial \boldsymbol{z}_q}{\partial \theta} &= -\sigma_1 \frac{\partial \boldsymbol{v}_\theta(\mathrm{sg}(\boldsymbol{x}_1^{(q)}), \sigma_1, \boldsymbol{c})}{\partial \theta}, \qquad q \geq 1. \end{aligned}$$

The loss averages the resulting reward gradients. Every target has nested depth zero, but only $\boldsymbol{z}_0$ equals the cached endpoint.

DRTune sets $j = 0$ and samples a sparse set t_{train} of equally spaced training steps together with an independent early-stop index $t_{\min}$. Define $\mathcal{A}_> := t_{\text{train}} \cap \{t_{\min} + 1, \ldots, N\}$ and $c := \mathbf{1}\{t_{\min} \in$

$t_{\text{train}}\}$. The state at the stop index is built from the surrogate velocities, $\boldsymbol{x}_{t_{\min}} = \boldsymbol{x}_N - \sum_{i>t_{\min}} h_i \tilde{\boldsymbol{v}}_i$, and the reward reads $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_{t_{\min}} - \sigma_{t_{\min}} \tilde{\boldsymbol{v}}_{t_{\min}}$, so

$$\frac{\partial \hat{\boldsymbol{x}}_0}{\partial \theta} = -\sum_{i \in \mathcal{A}_>} h_i \, \partial_\theta \boldsymbol{v}_i - c\, \sigma_{t_{\min}} \partial_\theta \boldsymbol{v}_{t_{\min}}, \qquad d_{\text{nest}} = 0. \tag{28}$$

The closing term appears only on coinciding draws and vanishes if that predictor is always detached. Every active velocity detaches its input, so $d_{\text{nest}} = 0$; Lemma 9 gives $d_0 \le \frac{L}{2}\sigma^2_{t_{\min}}$.

LeapAlign is the only prior method to engage both connectors. It is Corollary 16: $M = 1$ support per segment, both anchors active, $\rho \equiv 1$, and $\alpha \in (0, 1]$. Its coverage is $w_k^{\mathrm{L}} + w_j^{\mathrm{L}} = \sigma_k$ by Equation (25), its nested depth is 1, and Lemma 9 bounds both residuals, $d_j \le \frac{L}{2}(\sigma_k - \sigma_j)^2$ and $d_0 \le \frac{L}{2}\sigma_j^2$.

Remark 17 (Implementation-level backward rescaling). Only FlowBP-Lagrange rescales active supports, using $\rho_i = 1$ at anchors and $\rho_i = g_s$ otherwise; reconstruction variants use $\rho_i = 1$. These scales leave forward values unchanged, and all bounds use $W_{\mathcal{A}} = \sum_{i \in \mathcal{A}} |\rho_i w_i|$.

B Additional Experimental Details

B.1 Training configuration

Table 5 summarizes the common training configuration used in our experiments. All methods are trained with HPSv2.1 as the differentiable reward model and use the hinge reward loss

$$\mathcal{L}_{\text{reward}} = \mathrm{ReLU}(\lambda - r(\boldsymbol{x}_0, \boldsymbol{c})), \tag{29}$$

where the reward is evaluated at each method's clean target (with the fixed decoder implicit in r), $\boldsymbol{c}$ is the text prompt, and $\lambda = 0.55$. Following the LeapAlign setup, connector-based methods further apply trajectory-similarity weighting to this raw hinge loss. For a connector-based bridged surrogate, we measure the connector residuals at the bridge and clean endpoint,

$$d_j = \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|_2, \qquad d_0 = \|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\|_2, \tag{30}$$

where $\hat{\boldsymbol{x}}_j$ and $\hat{\boldsymbol{x}}_0$ are the surrogate predictions before the straight-through connectors. We then form

$$w_{\text{sim}} = \frac{1}{\max(d_j, \tau) + \max(d_0, \tau)}, \qquad \mathcal{L} = \mathrm{sg}(w_{\text{sim}})\, \mathcal{L}_{\text{reward}}. \tag{31}$$

Here $\tau > 0$ is a residual floor (clamp) on d_j and d_0, avoiding division by a near-zero denominator. The stop-gradient on w_{sim} makes d_j and d_0 act only as a scalar loss reweighting: trajectories whose surrogate connectors better match the cached rollout receive larger training weight, while the gradient-routing graph remains the one specified by the active set, integration weights, and bridge scale. For connector-based methods without an interior bridge, the same rule reduces to endpoint-deviation weighting with the d_j term omitted. Unless otherwise specified, we use loss scale 1.0, AdamW (Loshchilov & Hutter, 2019) with learning rate 1×10^{-5}, weight decay 10^{-4}, no learning-rate warmup, maximum gradient norm 1.0, EMA decay 0.995, and batch size 64. Every configuration, including base models, baselines, our variants, and the ablations, is run with three random seeds, and every number we report is the mean over those three runs. Online rollouts use 25 sampling steps at 512×512 resolution. We use classifier-free guidance scale 3.5 for `SD3.5-M` and 4.0 for the FLUX backbones. We train `SD3.5-M` for 250 iterations and both `FLUX.1-dev` and `FLUX.2-Klein-base` for 300 iterations. We enable gradient checkpointing, full-shard FSDP, TF32, EMA, and bf16 mixed precision. For non-CFG-distilled models, we detach the negative branch in classifier-free guidance.

Table 5: Backbone-specific training configuration. All runs use HPSv2.1 as the training reward and 25-step online rollouts at 512×512 resolution.

Backbone	GPUs	Iter.	CFG	Batch	LR	WD	EMA
`SD3.5-M`	8×H20	250	3.5	64	1×10^{-5}	10^{-4}	0.995
`FLUX.1-dev`	16×H20	300	4.0	64	1×10^{-5}	10^{-4}	0.995
`FLUX.2-Klein-base`	16×H20	300	4.0	64	1×10^{-5}	10^{-4}	0.995

B.2 TRAJECTORY SAMPLING

Connector-style leap methods draw two indices $0 < j < k \leq N$ per training step, and where those indices land on the trajectory affects optimization stability: an endpoint placed in the high-noise region makes a single velocity a poor summary of the interval it spans and inflates the variance of the reward gradient. We therefore confine the eligible indices to the clean end of the rollout through a step-window fraction $\omega \in (0, 1]$, which makes only the last $\lfloor \omega N \rfloor$ sampling steps eligible, so that $0 < j < k \leq \lfloor \omega N \rfloor$; k and j are then drawn uniformly at random from that range. Setting $\omega = 1$ leaves the whole trajectory eligible. The broader main-text convention is $N \geq k > j \geq 0$, where $j = 0$ denotes a single-segment surrogate without an interior bridge. For FlowBP-Bridge, only the split j is used and the noisy-side endpoint is fixed to N, consistent with its full Euler reconstruction. Unless otherwise stated, the nested-gradient scale is 0.3.

Surrogate supports and reconstruction. All our variants first perform a no-gradient rollout to cache trajectory states and velocities, and then re-forward only selected active velocities with gradients. For FlowBP-Lagrange, an interval from s to t (with $s > t$, so $\sigma_s > \sigma_t$) is approximated by a Lagrange connector with per-leap support set $\mathcal{S}$,

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_s - \sum_{i \in \mathcal{S}} w_i \tilde{\boldsymbol{v}}_i, \qquad w_i = \int_{\sigma_t}^{\sigma_s} \prod_{q \in \mathcal{S}, q \neq i} \frac{\sigma - \sigma_q}{\sigma_i - \sigma_q} \, d\sigma, \tag{32}$$

where $M = |\mathcal{S}|$ is the number of supports on that leap. Lemma 14 shows that these weights sum to the segment length $\sigma_s - \sigma_t$ but may be signed; the implementation and analysis retain their signs, while norm and coverage calculations use absolute values. For FlowBP-Sparse, the clean endpoint is reconstructed with Euler weights,

$$\boldsymbol{x}_0 = \boldsymbol{x}_N - \sum_{i=1}^{N} h_i \tilde{\boldsymbol{v}}_i, \qquad h_i = \sigma_i - \sigma_{i-1} > 0. \tag{33}$$

For FlowBP-Bridge, the same Euler reconstruction is applied separately on $(N \to j)$ and $(j \to 0)$. Active steps in FlowBP-Sparse and FlowBP-Bridge are drawn uniformly at random from their segment. In FlowBP-Lagrange, the extra supports that the high-order rule adds beyond the segment anchors would otherwise inflate the gradient norm, so we attenuate their backward contribution by a gradient-support scale $g_s \in [0, 1]$, applied through the stop-gradient identity

$$\tilde{\boldsymbol{v}}_i^{\rho} = \mathrm{sg}(\tilde{\boldsymbol{v}}_i) + \rho_i \big(\tilde{\boldsymbol{v}}_i - \mathrm{sg}(\tilde{\boldsymbol{v}}_i)\big), \qquad \rho_i = \begin{cases} 1, & i \text{ a segment anchor}, \\ g_s, & i \text{ a non-anchor active support}, \end{cases} \tag{34}$$

which leaves the forward trajectory unchanged. The reconstruction variants use $\rho_i = 1$ throughout. Every method therefore uses the effective backward weight $\beta_i = \rho_i w_i$ of Equation (5). The concrete hyperparameters are summarized in Table 6.

FLUX.2-specific implementation. For `FLUX.2-Klein-base`, we precompute Qwen3 text embeddings using a maximum sequence length of 512 and concatenate hidden states from layers 9, 18, and 27 as the text-conditioning representation.

B.3 EVALUATION

We evaluate HPDv2 on the 400-prompt test split with 50-step sampling and report average scores and gains over the base model, each averaged over the three seeds.

Seed variation. Table 7 reports the `SD3.5-M` means from Table 2 together with the standard deviation over multiple random seeds.

Compositional generation on GenEval. We evaluate compositional generation on `FLUX.1-dev` with GenEval (Ghosh et al., 2023), following its official protocol and reporting task accuracies together with the overall score from the official evaluator. Table 8 gives the breakdown. FlowBP-Lagrange lifts the overall score from 63.25 to 69.88 and is best on three of the six categories, while FlowBP-Bridge is second overall and best on spatial position, so the preference gains of Table 2 do not come at the cost of compositional prompt following. Representative samples are shown in Figure 2.

Table 6: Method-specific hyperparameters used for baselines and our variants. Common optimization, reward, rollout, and precision settings are reported in Table 5. Here, K denotes the number of active steps, α is the nested-gradient scale, g_s is the gradient-support scale, ω is the step-window fraction restricting the leap indices to the last $\lfloor \omega N \rfloor$ sampling steps, τ is the residual floor, and n is the number of noised samples in DRaFT-LV.

Method	Model	Hyperparameters
ReFL	All	Last-step window $N_{\text{last}} = 11$
DRaFT-LV	All	Noised samples $n = 2$
DRTune	All	Active train steps $K = 3$; early-stop ratio 0.4
LeapAlign	All	$\alpha = 0.3$; trajectory weighting $\tau = 0.1$
FlowBP-Lagrange	Shared	$M = 3$ supports per segment; $g_s = 0.25$; $\alpha = 0.1$; $\tau = 0.2$
	`SD3.5-M`	$\omega = 0.7$
	`FLUX.1-dev`	$\omega = 1.0$
	`FLUX.2-Klein-base`	$\omega = 0.7$
FlowBP-Sparse	All	Active steps sampled uniformly at random; $K = 4$
FlowBP-Bridge	Shared	Active steps sampled uniformly at random; $K = 4$
	`SD3.5-M`	$\alpha = 0.3$
	`FLUX.1-dev`	$\alpha = 0.1$
	`FLUX.2-Klein-base`	$\alpha = 1.0$

Table 7: Multi-seed results on `SD3.5-M`. We report mean $\pm$ std over random seeds.

Method	**HPSv2.1↑**	**PickScore↑**	**ImageReward↑**	**UR-Align↑**	**UR-IQ↑**
SD3.5-M					
ReFL	0.3872 ± 0.0006	23.4021 ± 0.01	1.3799 ± 0.0204	3.4779 ± 0.006	3.9464 ± 0.016
DRaFT-LV	0.3818 ± 0.0006	23.3506 ± 0.03	1.3673 ± 0.0298	$\underline{3.5072 \pm 0.02}$	3.9245 ± 0.019
DRTune	0.3870 ± 0.0022	23.2882 ± 0.03	1.3794 ± 0.0224	3.4562 ± 0.011	3.8869 ± 0.021
LeapAlign	0.3924 ± 0.0022	23.3142 ± 0.07	$\underline{1.4425 \pm 0.007}$	3.4854 ± 0.014	3.9030 ± 0.033
FlowBP-Sparse	0.3939 ± 0.0014	23.4104 ± 0.02	1.4299 ± 0.0086	3.4774 ± 0.021	$\mathbf{3.9690 \pm 0.012}$
FlowBP-Bridge	$\underline{0.3966 \pm 0.0003}$	$\underline{23.4130 \pm 0.01}$	1.4386 ± 0.0114	3.4712 ± 0.005	3.9481 ± 0.021
FlowBP-Lagrange	$\mathbf{0.3998 \pm 0.0019}$	$\mathbf{23.5040 \pm 0.05}$	$\mathbf{1.4538 \pm 0.002}$	$\mathbf{3.5127 \pm 0.003}$	$\underline{3.9680 \pm 0.017}$

Table 8: GenEval results on FLUX.1-dev, including the overall score and task-level accuracies. Δ denotes the improvement over the base model. Best and second-best results are shown in **bold** and underlined, respectively. Blue rows indicate our methods.

Method	**Overall**		**Single Obj.**		**Two Obj.**		**Count**		**Color**		**Pos**		**AttrB**	
	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)
FLUX.1-dev 12B														
`FLUX.1-dev`	63.25	—	97.81	—	76.52	—	70.31	—	76.86	—	17.75	—	40.25	—
ReFL	65.94	+2.69	97.50	−0.31	77.27	+0.75	71.88	+1.57	<u>78.99</u>	<u>+2.13</u>	22.25	+4.50	47.75	+7.50
DRaFT-LV	68.31	+5.06	98.13	+0.32	**86.36**	**+9.84**	**75.00**	**+4.69**	77.13	+0.27	23.25	+5.50	50.00	+9.75
DRTune	66.78	+3.53	<u>98.75</u>	<u>+0.94</u>	80.56	+4.04	72.81	+2.50	76.33	−0.53	20.75	+3.00	51.50	+11.25
LeapAlign	67.44	+4.19	98.44	+0.63	84.09	+7.57	72.19	+1.88	78.19	+1.33	22.25	+4.50	49.50	+9.25
FlowBP-Sparse	68.19	+4.94	<u>98.75</u>	<u>+0.94</u>	84.85	+8.33	<u>73.13</u>	<u>+2.82</u>	78.19	+1.33	22.00	+4.25	52.25	+12.00
FlowBP-Bridge	<u>68.75</u>	<u>+5.50</u>	96.88	−0.93	<u>85.10</u>	<u>+8.58</u>	70.94	+0.63	76.86	+0.00	**27.25**	**+9.50**	<u>55.50</u>	<u>+15.25</u>
FlowBP-Lagrange	**69.88**	**+6.63**	**99.06**	**+1.25**	<u>85.10</u>	<u>+8.58</u>	71.88	+1.57	**79.52**	**+2.66**	<u>23.75</u>	<u>+6.00</u>	**60.00**	**+19.75**

B.4 HYPERPARAMETER SWEEPS

Figure 5 reports the sensitivity of the reconstruction-based variants on `FLUX.1-dev`: the nested-gradient scale α and the active-step budget K for FlowBP-Bridge, and the active-step budget K for FlowBP-Sparse. The corresponding gradient expressions are derived in Appendices A.5 and A.6.

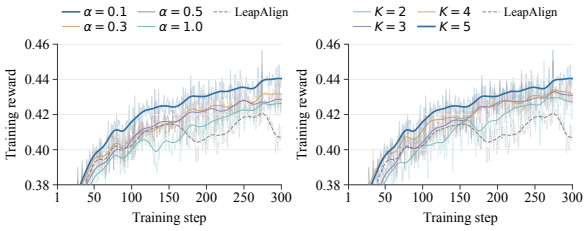


Figure 5: Ablation of FlowBP-Bridge. Left: effect of the nested-gradient scale. Right: effect of active-step budget K.

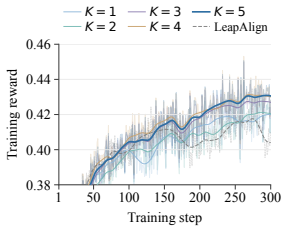


Figure 6: Ablation of FlowBP-Sparse active-step budget K.

B.5 EVALUATION DYNAMICS

In addition to final scores, Figure 7 reports PickScore, ImageReward, and HPSv2.1 over training for all three backbones, showing whether gains persist across checkpoints.

B.6 A STRONGER REWARD MODEL

HPSv2.1 is a CLIP-scale scorer. We repeat the study on `SD3.5-M` against HPSv3++ (Liu et al., 2026), a Qwen3-VL-8B preference model, to ask which surrogate produces the better gradient under a modern reward.

Gradient clipping is active for every method at every step of these runs, normalizing the gradient supplied to the optimizer to the clipping threshold up to numerical tolerance. This does not equalize AdamW parameter-update norms: momentum, adaptive preconditioning, and weight decay still affect each update. The scores therefore compare the complete training procedures, rather than isolating gradient direction or directly measuring gradient fidelity.

One setting has to change with the reward. The hinge $\mathrm{ReLU}(\lambda - \mu)$ needs λ above the reward scale, and HPSv3++ already scores $\mu \approx 6.8$ on untrained samples, so the $\lambda = 0.55$ of the standard recipe would leave the loss identically zero. We use $\lambda = 20$, above every observed μ. The constant cancels out of the raw hinge-loss gradient, giving $-\nabla\mu$ on these samples; trajectory weighting and gradient clipping still apply. Table 9 gives the full configuration alongside the standard one, and Table 10 the per-method settings.

The endpoint intervention remains beneficial in-domain. Moving the reward onto the sampled endpoint (row B), with ReFL's active step and base weight unchanged, lifts the in-domain score by 0.2966 with mixed held-out changes. Row C adds $+0.3715$ in-domain, but changes the active set, integration weights, and sampling window (from the last 11 steps to the full trajectory) together (Table 11). In the method comparison, the three FlowBP variants reach the highest in-domain scores (Table 12), while improving both held-out metrics over the base model. DRTune gains in-domain but ends 0.41 ImageReward *below* the base model, and LeapAlign ends only 0.02 above it, whereas every FlowBP variant retains at least $+0.18$.

The bridge configuration does not consistently improve scores. On `FLUX.1-dev` under HPSv2.1 it improved all five metrics over the sparse configuration (Table 3, row D); here row D sits 0.0904 below row C in-domain and 0.0323 below it on ImageReward, and removing the coupling (α=0) recovers most of the in-domain gap while giving the better held-out HPSv2.1. The main text already reports that the deployed scale has no backbone-independent value, ranging from α=0.1 on `FLUX.1-dev` to α=1.0 on `FLUX.2-Klein-base` (Table 6). This run adds the reward model as a second factor that moves it, and shows that the useful range extends down to α=0. Tuned coupling can yield a small gain, but its benefit is less stable across settings. We therefore recommend uncoupled FlowBP-Sparse

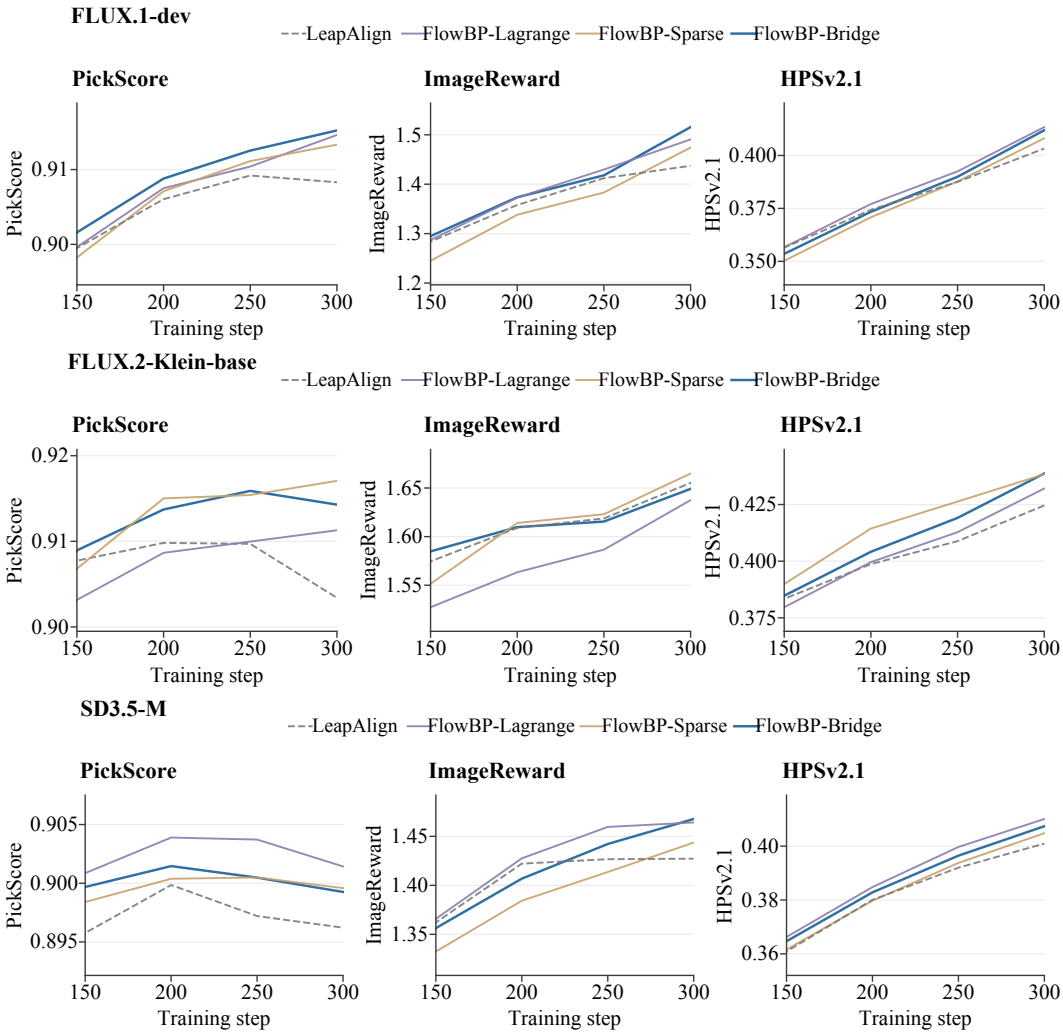


Figure 7: Evaluation dynamics on the HPDv2 test split for `FLUX.1-dev`, `FLUX.2-Klein-base`, and `SD3.5-M`. Each row reports PickScore, ImageReward, and HPSv2.1 over training, with PickScore normalized by 26.

with $K = 3$ for routine use, treating bridge coupling as an optional refinement when further tuning is justified. The comparison runs above retain their reported $K = 4$ settings.

The Lagrange configuration remains competitive. Row E recovers most of the in-domain score that row D gives up and returns the highest held-out ImageReward in Table 11, so the multi-support connector remains a viable alternative to exact replay under this reward as well. Its eligible window trades the two directions against each other in the expected way: widening it to the full trajectory (ω=1.0) produces the highest in-domain score and the highest held-out HPSv2.1 in the table, and gives back ImageReward to do so.

Table 9: Configuration of the HPSv3++ study against the standard recipe. Both train `SD3.5-M` with the same optimizer, batch size, and online rollout; reward, hinge threshold, GPU count, and training duration differ.

Setting	**Standard recipe**	**HPSv3++ study**
Training reward	HPSv2.1 (CLIP-scale)	HPSv3++ (Qwen3-VL-8B)
Reward at initialization	≈ 0.29	≈ 6.8
Hinge threshold λ	0.55	20
Gradient clipping	1.0, rarely binding	1.0, binding every step
Backbone	`SD3.5-M`	`SD3.5-M`
GPUs	8×H20	16×H20
Iterations	250	300
Batch size	64	64
Learning rate / WD	1×10^{-5} / 10^{-4}	1×10^{-5} / 10^{-4}
EMA decay	0.995	0.995
CFG scale	3.5	3.5
Online rollout	25 steps, 512×512	25 steps, 512×512
Evaluation	400 HPDv2 test prompts, 50-step sampling, CFG 3.5	

Table 10: Method-specific hyperparameters for the HPSv3++ study on `SD3.5-M`. Shared optimization, reward and rollout settings are in Table 9. Notation follows Table 6: K is the number of active steps, α the nested-gradient scale, g_s the gradient-support scale, ω the step-window fraction, τ the residual floor, and n the number of noised samples in DRaFT-LV. The lower block lists the controlled variants, each of which changes exactly one entry of its reference row.

Method	Row	Hyperparameters
ReFL	A	Last-step window $N_{\text{last}} = 11$; reward on the one-step clean estimate
ReFL + endpoint connector	B	Same active step and σ_τ weight as ReFL; reward on the sampled endpoint through a straight-through connector
FlowBP-Sparse	C	Active steps sampled uniformly at random; $K = 4$
FlowBP-Bridge	D	Active steps sampled uniformly at random; $K = 4$; $\alpha = 0.3$
FlowBP-Lagrange	E	$M = 3$ supports per segment; $g_s = 0.25$; $\alpha = 0.1$; $\tau = 0.2$; $\omega = 0.7$
DRaFT-LV	—	Noised samples $n = 2$
DRTune	—	Active train steps $K = 3$; early-stop ratio 0.4
LeapAlign	—	$\alpha = 0.3$; trajectory weighting $\tau = 0.1$
Controlled variants		
FlowBP-Bridge, uncoupled	↪ D	As row D with $\alpha = 0$
FlowBP-Lagrange, full trajectory	↪ E	As row E with $\omega = 1.0$

Table 11: **HPSv3++** ablation on `SD3.5-M`. Labels match Figure 8. Δ is relative to the preceding A–E row; D_0 and E_1 are compared with D and E, respectively.

Configuration	In-Domain		Out-of-Domain			
	HPSv3++		ImageReward		HPSv2.1	
	Score↑	Δ	Score↑	Δ	Score↑	Δ
`SD3.5-M`	7.0457	—	0.9643	—	0.2882	—
A ReFL	15.3967	—	1.1595	—	0.3440	—
B + endpoint	15.6933	+0.2966	1.1610	+0.0015	0.3413	−0.0027
C Sparse	<u>16.0648</u>	+0.3715	<u>1.1803</u>	+0.0193	0.3449	+0.0036
D Bridge	15.9744	−0.0904	1.1479	−0.0323	0.3443	−0.0005
D_0 $\alpha = 0$	16.0362	+0.0618	1.1191	−0.0289	<u>0.3453</u>	+0.0010
E Lagrange	16.0423	+0.0679	**1.1808**	+0.0328	0.3452	+0.0009
E_1 $\omega = 1$	**16.0821**	+0.0398	1.1472	−0.0335	**0.3463**	+0.0011

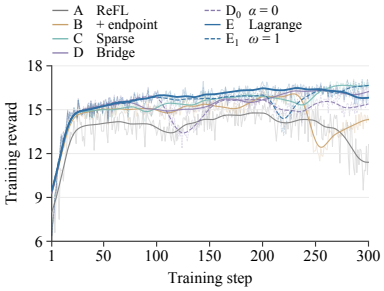


Figure 8: HPSv3++ training reward for the rows of Table 11. A controlled variant shares the colour of its reference and is dashed. Light traces are raw values, solid lines Gaussian-smoothed.

Table 12: Method comparison on `SD3.5-M` with **HPSv3++** as the differentiable training reward. Δ is the gain over the untrained base model. Best and second-best are **bold** and <u>underlined</u>; blue rows are our variants.

Method	In-Domain		Out-of-Domain			
	HPSv3++		ImageReward		HPSv2.1	
	Score↑	Δ(↑)	Score↑	Δ(↑)	Score↑	Δ(↑)
`SD3.5-M` (no post-training)	7.0457	—	0.9643	—	0.2882	—
ReFL	15.3967	+8.3510	1.1595	+0.1952	0.3440	+0.0559
DRaFT-LV	15.6463	+8.6007	**1.2149**	+0.2507	0.3431	+0.0549
DRTune	14.3338	+7.2882	0.5530	−0.4113	0.3239	+0.0357
LeapAlign	15.7524	+8.7068	0.9834	+0.0191	0.3422	+0.0540
FlowBP-Sparse	**16.0648**	+9.0192	1.1803	+0.2160	<u>0.3449</u>	+0.0567
FlowBP-Bridge	15.9744	+8.9287	1.1479	+0.1837	0.3443	+0.0561
FlowBP-Lagrange	<u>16.0423</u>	+8.9966	<u>1.1808</u>	+0.2165	**0.3452**	+0.0570

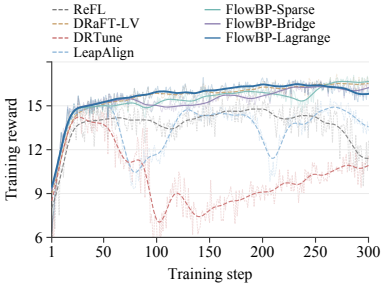


Figure 9: HPSv3++ training reward for the methods of Table 12. Prior baselines are dashed and our variants solid. The three FlowBP variants rise steadily and remain near their maximum.

C ALGORITHMIC TEMPLATES FOR REWARD-GRADIENT UPDATES

This appendix specifies the reward-gradient update graphs used by the methods discussed in the main text. Each template identifies the reward input, the active velocity evaluations, and the surrogate path through which gradients are propagated.

We present two templates. Short clean-estimate updates query the reward on one or more model-implied predictions, whereas cached-rollout surrogate updates first sample a full trajectory without gradients and then construct a sparse backward graph on top of the cached states. Together, the two templates make the gradient-routing differences between the compared methods explicit.

C.1 SHORT CLEAN-ESTIMATE REWARD UPDATES

ReFL, DRTune, and DRaFT-LV can be written as short clean-estimate updates. They differ in the stop index, rollout-active velocity calls, whether the closing predictor is active, and the number of reward targets. The estimates equal conditional posterior means only for a population-optimal rectified-flow predictor. In DRTune the stop index is sampled independently of the equally spaced training steps, so the closing predictor carries a gradient only on coinciding draws. DRaFT-LV additionally uses fresh active final-step predictors at detached re-noised inputs. The shared template is given in Algorithm 1.

Algorithm 1: Unified Short Clean-Estimate Updates

Input: Velocity model $\boldsymbol{v}_\theta$; reward function $r(\cdot,\cdot)$; prompt distribution p_c; hinge threshold λ; noise schedule $\{\sigma_i\}_{i=0}^{N}$; method flag $m \in \{\text{ReFL}, \text{DRTune}, \text{DRaFT-LV}\}$.

Output: Updated model parameters θ.

Unified rollout graph

Sample $\boldsymbol{c} \sim p_c$, $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Choose a stop index e, rollout-active set $\mathcal{A}_> \subseteq \{e+1, \dots, N\}$, closing-call flag $c_e \in \{0,1\}$, and reward-target set $\mathcal{Q}$:

ReFL: $(e, \mathcal{A}_>, c_e, \mathcal{Q}) = (\tau, \emptyset, 1, \{0\})$, where $\tau \in \mathcal{T}_{\text{tail}}$.

DRTune: $e = t_{\min}$, $\mathcal{A}_> = t_{\text{train}} \cap \{e+1, \dots, N\}$, $c_e = \mathbf{1}\{e \in t_{\text{train}}\}$, and $\mathcal{Q} = \{0\}$.

DRaFT-LV: $(e, \mathcal{A}_>, c_e, \mathcal{Q}) = (1, \emptyset, 1, \{0, \dots, n\})$.

For all executed steps, use

$$\tilde{\boldsymbol{v}}_i = \begin{cases} \boldsymbol{v}_\theta(\text{sg}(\boldsymbol{x}_i), \sigma_i, \boldsymbol{c}), & i \in \mathcal{A}_>, \\ \text{sg}\big(\boldsymbol{v}_\theta(\text{sg}(\boldsymbol{x}_i), \sigma_i, \boldsymbol{c})\big), & i \notin \mathcal{A}_>, \end{cases} \qquad \boldsymbol{x}_{i-1} = \boldsymbol{x}_i - h_i\, \tilde{\boldsymbol{v}}_i, \qquad h_i = \sigma_i - \sigma_{i-1} > 0.$$

Unified reward targets

Define the closing velocity and direct clean target

$$\bar{\boldsymbol{v}}_e = \begin{cases} \boldsymbol{v}_\theta(\text{sg}(\boldsymbol{x}_e), \sigma_e, \boldsymbol{c}), & c_e = 1, \\ \text{sg}\big(\boldsymbol{v}_\theta(\text{sg}(\boldsymbol{x}_e), \sigma_e, \boldsymbol{c})\big), & c_e = 0, \end{cases} \qquad \boldsymbol{z}_0 = \boldsymbol{x}_e - \sigma_e\, \bar{\boldsymbol{v}}_e.$$

For DRaFT-LV only, keep $\boldsymbol{z}_0$ as the first reward target ($q = 0$). Then construct n additional targets by re-noising its detached value and applying a fresh active final-step predictor:

$$\boldsymbol{x}_1^{(q)} = (1-\sigma_1)\,\text{sg}(\boldsymbol{z}_0) + \sigma_1 \boldsymbol{\epsilon}_q, \qquad \boldsymbol{z}_q = \boldsymbol{x}_1^{(q)} - \sigma_1\, \boldsymbol{v}_\theta(\text{sg}(\boldsymbol{x}_1^{(q)}), \sigma_1, \boldsymbol{c}), \quad q = 1, \dots, n.$$

Reward update

Compute $\mathcal{L} = |\mathcal{Q}|^{-1} \sum_{q \in \mathcal{Q}} \text{ReLU}\big(\lambda - r(\boldsymbol{z}_q, \boldsymbol{c})\big)$, where the single-target cases use $\mathcal{Q} = \{0\}$ and therefore only the $\boldsymbol{z}_0$ term.

Update $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}$.

return θ.

C.2 Cached-Rollout Surrogate Updates

LeapAlign and the proposed FlowBP variants use a different graph construction. They first generate and cache a complete rollout without storing activations, and then expose only selected velocity evaluations to autograd. Connector-based variants attach sparse predictions to cached states with straight-through connectors, whereas reconstruction-based variants replay Euler updates with cached inactive velocities. Algorithm 2 summarizes this family of update graphs.

Algorithm 2: Unified Surrogate-Trajectory Gradient Methods

Input: Velocity model $\boldsymbol{v}_\theta$; scalar reward $r(\boldsymbol{x}_0, \boldsymbol{c})$; prompt distribution p_c; hinge threshold λ; noise schedule $\{\sigma_i\}_{i=0}^N$; bridge scale α; support scale g_s; residual floor τ; optimizer step size γ; method flag $m \in \{\text{LeapAlign}, \text{FlowBP-Lagrange}, \text{FlowBP-Sparse}, \text{FlowBP-Bridge}\}$.

Output: Updated model parameters θ.

Unified cached rollout

Sample $\boldsymbol{c} \sim p_c$, $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Run a no-gradient rollout and cache $\{\boldsymbol{x}_i\}_{i=0}^N$ and $\{(\boldsymbol{v}_i, \sigma_i)\}_{i=1}^N$.

Choose indices, active sets $\mathcal{A} = \mathcal{A}_{\text{pre}} \cup \mathcal{A}_{\text{post}}$, and weights $\{w_i\}$:

LeapAlign: choose $0 < j < k \le N$, $\mathcal{A}_{\text{pre}} = \{k\}$, $\mathcal{A}_{\text{post}} = \{j\}$, $w_k = \sigma_k - \sigma_j$, $w_j = \sigma_j$.

FlowBP-Lagrange: choose $0 < j < k \le N$, Lagrange supports $\mathcal{S}_{\text{pre}}$ and $\mathcal{S}_{\text{post}}$ anchored at k and j, active sets $\mathcal{A}_{\text{pre}} \subseteq \mathcal{S}_{\text{pre}}$, $\mathcal{A}_{\text{post}} \subseteq \mathcal{S}_{\text{post}}$ with $k \in \mathcal{A}_{\text{pre}}$, $j \in \mathcal{A}_{\text{post}}$ and $|\mathcal{A}_{\text{pre}} \cup \mathcal{A}_{\text{post}}| \le 2M$ for M supports per segment, and $w_i = w_i^{\text{L}}$.

FlowBP-Sparse: $j = 0$, $\mathcal{A}_{\text{post}} = \emptyset$, $\mathcal{A} = \mathcal{A}_{\text{pre}} \subseteq \{1, \dots, N\}$, $w_i = h_i$.

FlowBP-Bridge: choose $j \in \{1, \dots, N-1\}$, $\mathcal{A}_{\text{pre}} \subseteq \{j+1, \dots, N\}$, $\mathcal{A}_{\text{post}} \subseteq \{1, \dots, j\}$ with $j \in \mathcal{A}_{\text{post}}$, $w_i = h_i$.

Define the non-bridge base velocities $\tilde{\boldsymbol{v}}_i$ ($i \neq j$) by the second and third branches of Equation (4); defer the bridge call at $i = j$ until the differentiable split latent has been constructed.

Set $\rho_i = g_s$ for non-anchor active FlowBP-Lagrange supports and $\rho_i = 1$ for anchors and for the active supports of the other methods, and let

$$\tilde{\boldsymbol{v}}_i^\rho = \begin{cases} \text{sg}(\tilde{\boldsymbol{v}}_i) + \rho_i(\tilde{\boldsymbol{v}}_i - \text{sg}(\tilde{\boldsymbol{v}}_i)), & i \in \mathcal{A},\ i \neq j, \\ \text{sg}(\boldsymbol{v}_i), & i \notin \mathcal{A}. \end{cases}$$

Unified surrogate targets

For connector-based bridged methods, attach the pre-segment surrogate to the cached bridge state:

$$\hat{\boldsymbol{x}}_j = \begin{cases} \boldsymbol{x}_k - w_k\, \tilde{\boldsymbol{v}}_k^\rho, & \text{LeapAlign}, \\ \boldsymbol{x}_k - \sum_{i \in \mathcal{S}_{\text{pre}}} w_i^{\text{L}}\, \tilde{\boldsymbol{v}}_i^\rho, & \text{FlowBP-Lagrange}, \end{cases} \qquad \boldsymbol{x}_j = \hat{\boldsymbol{x}}_j + \text{sg}(\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j).$$

For FlowBP-Bridge, reconstruct the bridge state by Euler reconstruction

$$\boldsymbol{x}_j = \boldsymbol{x}_N - \sum_{i=j+1}^{N} h_i\, \tilde{\boldsymbol{v}}_i^\rho.$$

For a bridged method, now evaluate

$$\tilde{\boldsymbol{v}}_j = \boldsymbol{v}_\theta(\alpha \boldsymbol{x}_j + (1-\alpha)\,\text{sg}(\boldsymbol{x}_j), \sigma_j, \boldsymbol{c}), \qquad \tilde{\boldsymbol{v}}_j^\rho = \text{sg}(\tilde{\boldsymbol{v}}_j) + \rho_j(\tilde{\boldsymbol{v}}_j - \text{sg}(\tilde{\boldsymbol{v}}_j)).$$

This ordering makes the bridge call depend on the just-connected or reconstructed $\boldsymbol{x}_j$ and retains the nested term of Equation (8).

Define the clean endpoint

$$\boldsymbol{x}_0 = \begin{cases} \hat{\boldsymbol{x}}_0 + \text{sg}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0), & \hat{\boldsymbol{x}}_0 = \boldsymbol{x}_j - w_j\, \tilde{\boldsymbol{v}}_j^\rho, & \text{LeapAlign}, \\ \hat{\boldsymbol{x}}_0 + \text{sg}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0), & \hat{\boldsymbol{x}}_0 = \boldsymbol{x}_j - \sum_{i \in \mathcal{S}_{\text{post}}} w_i^{\text{L}}\, \tilde{\boldsymbol{v}}_i^\rho, & \text{FlowBP-Lagrange}, \\ \boldsymbol{x}_N - \sum_{i=1}^{N} h_i\, \tilde{\boldsymbol{v}}_i^\rho, & & \text{FlowBP-Sparse}, \\ \boldsymbol{x}_j - \sum_{i=1}^{j} h_i\, \tilde{\boldsymbol{v}}_i^\rho, & & \text{FlowBP-Bridge}. \end{cases}$$

Reward update

Let $\mathcal{L}_{\text{reward}} = \text{ReLU}(\lambda - r(\boldsymbol{x}_0, \boldsymbol{c}))$. For LeapAlign and FlowBP-Lagrange, compute

$$w_{\text{sim}} = \frac{1}{\max(d_j, \tau) + \max(d_0, \tau)}, \qquad \mathcal{L} = \text{sg}(w_{\text{sim}})\mathcal{L}_{\text{reward}}.$$

For reconstruction methods, set $\mathcal{L} = \mathcal{L}_{\text{reward}}$.

Update $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}$ through active calls only.

return θ.

D Additional Qualitative Results

We provide additional qualitative examples generated from HPDv2 test prompts for `SD3.5-M`, `FLUX.1-dev`, and `FLUX.2-Klein-base` in Figures 10 to 12, respectively.

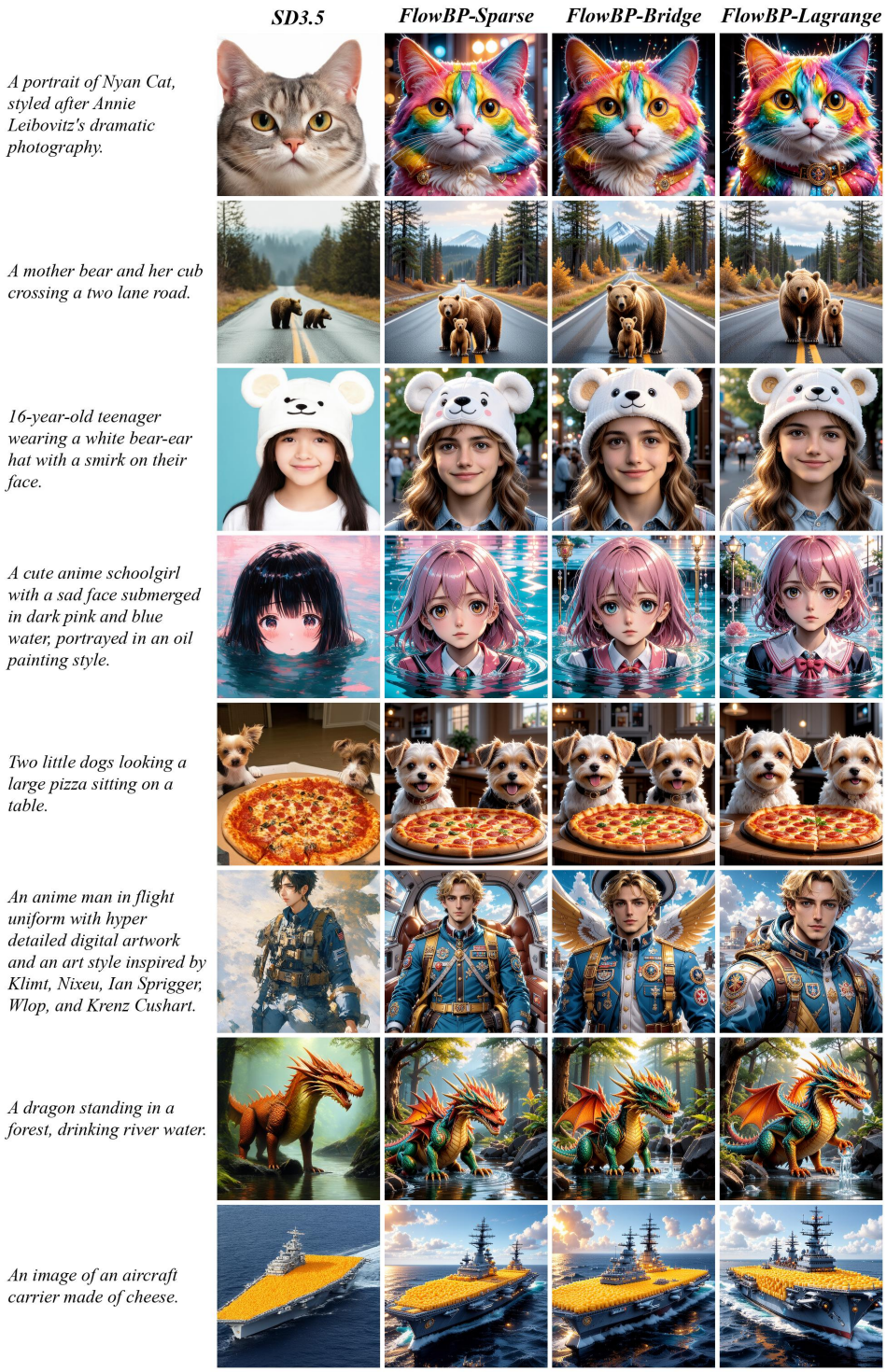


Figure 10: Additional qualitative results on `SD3.5-M` using prompts from the HPDv2 test split.



Figure 11: Additional qualitative results on `FLUX.1-dev` using prompts from the HPDv2 test split.

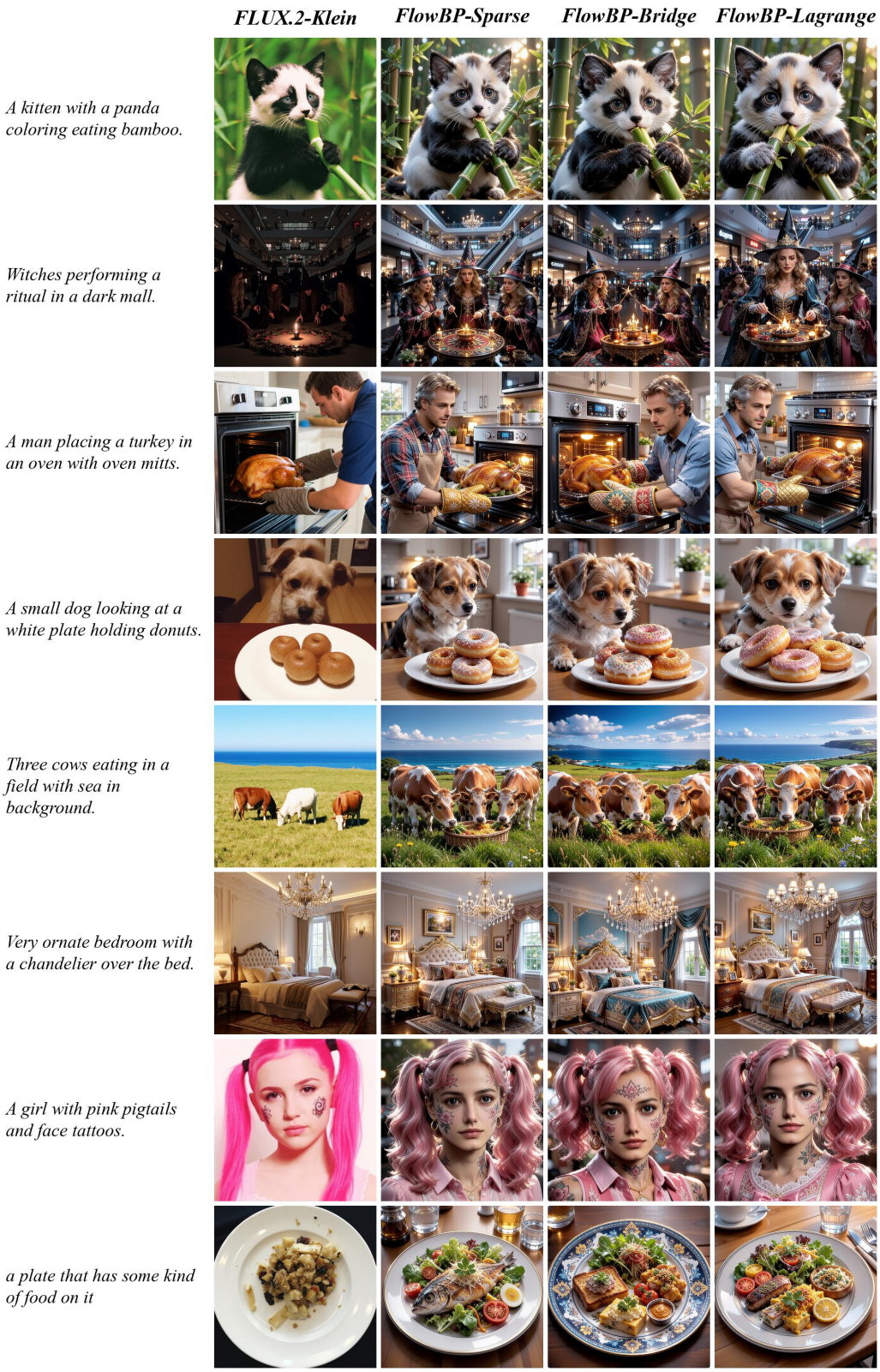


Figure 12: Additional qualitative results on `FLUX.2-Klein-base` using prompts from the HPDv2 test split.