MEANFLOWADVANTAGE: STABLE REWARD FINE-TUNING FOR FEW-STEP AVERAGE-VELOCITY GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

MeanFlow enables efficient few-step generation by predicting interval-average velocities, but this representation creates a mismatch for reward fine-tuning: existing advantage-based objectives are typically defined on instantaneous velocities or equivalent x_0-space predictions, whereas inference directly uses the learned average-velocity map. We introduce **MeanFlowAdvantage**, a signed advantage-weighted least-squares objective for average-velocity generators. Our key construction uses a shared, detached MeanFlow derivative correction to express the reward objective in prediction space while making rollout and reference regularization exact penalties on the average-velocity network deployed at inference. The resulting formulation preserves MeanFlow's native few-step sampler and provides a direct mechanism for transferring reward improvements to the deployed flow map. On SD3.5-Medium, MeanFlowAdvantage improves all eight reported metrics over the matched four-step MeanFlowNFT baseline and, with only four NFEs, matches or exceeds the 40-step DiffusionNFT baseline on six of eight metrics. The same objective also transfers to DNA promoter design, where it supports both teacher-free on-policy RL for a generator defined on a manifold and teacher-guided reward-graded distillation, with the latter yielding the lowest one-step Sei profile MSE among the compared configurations.

1 INTRODUCTION

Few-step flow matching changes the representation that post-training must optimize. In standard flow matching, the model predicts an instantaneous velocity whose numerical integration produces samples (Lipman et al., 2023; Liu et al., 2023; Albergo et al., 2023). MeanFlow (Geng et al., 2025; Gu et al., 2026), in contrast, predicts an *interval-average* velocity whose learned flow map can traverse a finite time interval in a single network evaluation. This representation is precisely what enables one-step and few-step sampling. However, it also creates a mismatch for reward fine-tuning: existing advantage-based objectives are typically formulated for instantaneous velocities or equivalent prediction spaces, whereas MeanFlow inference directly executes the learned average-velocity map. A useful reward objective should therefore improve sample quality without losing alignment with the representation that makes few-step generation possible.

Existing reinforcement learning (RL) approaches do not directly resolve this mismatch. Reverse-process methods such as DDPO, DPOK, and Flow-GRPO optimize stochastic denoising trajectories (Black et al., 2024; Fan et al., 2023; Liu et al., 2025), while forward-process methods incorporate reward signals into prediction objectives. DiffusionNFT uses implicit positive and negative policies (Zheng et al., 2026), Advantage Weighted Matching reweights a matching surrogate (Xue et al., 2025), and AdvantageFlow applies signed advantage-weighted least squares with rollout regularization (Kveton et al., 2026). These formulations naturally operate on instantaneous representations. Simply replacing such a predictor with a MeanFlow output is not equivalent: an interval-average velocity represents the transport over an entire interval rather than the instantaneous direction at one time point. Moreover, an induced predictor alone does not ensure that rollout or reference regularization constrains the positive-length average-velocity maps used by the sampler.

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.

Recent forward-process work has shown that MeanFlow can support reward fine-tuning through an induced instantaneous predictor (Huang et al., 2026b). This raises two questions: how can signed advantages be used to stably improve the finite-interval MeanFlow map, and can the same reward-based formulation also distill improved behavior from multi-step teachers into few-step MeanFlow models?

We introduce **MeanFlowAdvantage** (MFA), a signed advantage-weighted least-squares objective that provides a unified solution to both problems. MFA applies reward supervision in prediction space while regularizing the underlying average-velocity model against both the rollout policy and a frozen reference. A shared, stop-gradient correction makes these regularization terms reduce to direct differences between the corresponding MeanFlow velocity predictions, avoiding interference between reward fitting and model anchoring. This same formulation naturally extends from on-policy reward fine-tuning to reward-guided distillation, allowing few-step MeanFlow students to learn from improved multi-step teacher trajectories. Our contributions are threefold:

- **Reward fine-tuning aligned with average-velocity generation.** We formulate signed advantage optimization directly for finite-interval MeanFlow maps, while preserving the induced prediction space needed for reward supervision. This yields a representation-aligned objective that regularizes the average-velocity predictions used for generation rather than only the induced instantaneous boundary predictor.
- **State-of-the-art few-step performance in our evaluation.** On SD3.5-Medium, MFA achieves the strongest four-NFE results among the compared MeanFlow reward-finetuning methods, improving all eight reported metrics over MeanFlowNFT. With only four NFEs, it also matches or exceeds the 40-NFE DiffusionNFT baseline on six of eight metrics.
- **One objective for both on-policy RL and distillation.** On DNA promoters (Appendix E), the same loss runs as teacher-free RL on a manifold generator and, with a reward-guided teacher in place of the rollout, transfers graded teacher improvements into a one-step student, achieving the best Sei profile error among the compared configurations.

2 PRELIMINARIES

We use the image convention $t = 0$ for data and $t = 1$ for noise. The earlier interval endpoint is $s \leq t$, and c denotes conditioning. We omit c from the notation when clear from context, and distinguish average velocity u, instantaneous velocity v, and a training-time induced predictor V. The promoter implementation uses the opposite time convention, stated separately in Section E.

Flow matching Flow matching (Lipman et al., 2023) learns a continuous-time transport from a simple noise distribution to the conditional data distribution $q(x_0 \mid c)$. It represents this transport through a time-dependent velocity field whose probability-flow ODE is integrated from noise at $t = 1$ to data at $t = 0$ at inference time. Rectified flow specializes this construction to a linear interpolation between data and noise. For $x_0 \sim q(\cdot \mid c)$ and independent $\epsilon \sim \mathcal{N}(0, I)$, it defines

$$x_t = (1-t)x_0 + t\epsilon, \qquad v_t = \epsilon - x_0, \qquad \mathcal{L}_{\mathrm{FM}} = \mathbb{E}\|v_\theta(x_t, t) - v_t\|_2^2. \tag{1}$$

The population minimizer of this squared loss is the marginal instantaneous velocity $v(x_t, t, c) = \mathbb{E}[v_t \mid x_t, t, c]$. For rectified flow (Liu et al., 2023), the same instantaneous-velocity prediction can equivalently be expressed in x_0-space as $f_\theta = x_t - tv_\theta$, while $x_0 = x_t - tv_t$. Therefore, $f_\theta - x_0 = -t(v_\theta - v_t)$ and $\|f_\theta - x_0\|_2^2 = t^2\|v_\theta - v_t\|_2^2$. This prediction-space representation provides a convenient form for incorporating reward signals into forward-process objectives.

AdvantageFlow Building on this representation, AdvantageFlow (Kveton et al., 2026) incorporates signed reward information through a quadratic objective on the x_0-space predictions of the learner, rollout model, and frozen reference:

$$\ell^{\mathrm{AF}} = A\|f_\theta - x_0\|_2^2 + \gamma\|f_\theta - f_{\mathrm{old}}\|_2^2 + \lambda\|f_\theta - f_{\mathrm{ref}}\|_2^2. \tag{2}$$

Here A is a signed advantage and $\gamma, \lambda \geq 0$. The loss is strictly convex in the prediction when $A + \gamma + \lambda > 0$, even when $A < 0$. For standard flow models, this objective acts naturally on a prediction obtained from the instantaneous velocity. MeanFlow, however, deploys an interval-average velocity, so reward fine-tuning must additionally account for the finite-interval map used by its sampler.

MeanFlow and reward fine-tuning Along a trajectory $\dot{x}_\tau = v(x_\tau, \tau)$, MeanFlow represents transport over a finite interval $[s, t]$ through the average velocity

$$u(x_t, s, t) = \frac{1}{t-s} \int_s^t v(x_\tau, \tau)\, \mathrm{d}\tau, \qquad \lim_{s \to t} u(x_t, s, t) = v(x_t, t). \tag{3}$$

Holding s fixed and differentiating along the trajectory yields the MeanFlow identity

$$u + (t-s)\big[\partial_t u + (\partial_x u)v\big] = v. \tag{4}$$

For an exact average velocity, $x_s = x_t - (t-s)u(x_t, s, t)$ is the exact flow map; replacing u by u_θ gives the learned sampler $x_s = x_t - (t-s)u_\theta(x_t, s, t)$. Thus, unlike standard flow matching, MeanFlow inference directly uses the average-velocity maps predicted by u_θ. Reward fine-tuning should therefore remain aligned with these deployed maps. MeanFlowNFT uses the MeanFlow identity to construct an instantaneous predictor V from u and applies an NFT objective to V (Huang et al., 2026b). We use the same induction with the signed quadratic in equation 2, while regularizing the corresponding average-velocity maps directly; Section A.4 makes the connection to NFT explicit.

3 MeanFlowAdvantage

The preliminaries establish AdvantageFlow in prediction space and MeanFlow in average-velocity space. MeanFlowAdvantage connects these representations while keeping optimization aligned with the finite-interval maps deployed by the sampler. We first derive a shared MeanFlow correction that links the average velocity u to the prediction-space objective, and show that the resulting quadratic is exactly an advantage-weighted MeanFlow regression with rollout and reference anchors on the deployed maps. We then specify the samples, advantages, and training intervals used for optimization, followed by an analysis of the resulting objective and its fixed point.

3.1 Linking average velocity to prediction space

The naive substitution $x_t - t\, u_\theta(x_t, s, t)$ does not give the corresponding x_0-space prediction when $s < t$. In rectified flow, $x_t - tv$ depends on the instantaneous velocity at time t, whereas u averages velocity over the interval $[s, t]$. The MeanFlow identity equation 4 relates the two as $v = u + (t-s)\,\mathrm{D}u$, where $\mathrm{D}u = \partial_t u + (\partial_x u)v$ denotes the rate of change of u along the trajectory. This correction therefore links the average-velocity representation to the prediction space used by AdvantageFlow. Given a rollout sample x_0, we draw noise ϵ and an interval $s \le t$, form x_t by equation 1, and set $v_t = \epsilon - x_0$. With finite-difference step size $\delta > 0$, we estimate the correction once from the rollout network along the sampled direction v_t:

$$d = \frac{u_{\mathrm{old}}(x_t + \delta v_t, s, t+\delta) - u_{\mathrm{old}}(x_t - \delta v_t, s, t-\delta)}{2\delta} = \partial_t u_{\mathrm{old}} + (\partial_x u_{\mathrm{old}})\, v_t + O(\delta^2). \tag{5}$$

A Jacobian-vector product computes the same quantity without the $O(\delta^2)$ error. Appendix B gives the finite-difference implementation and boundary handling. We detach d, with $\mathrm{sg}[\cdot]$ denoting stop-gradient, and share the same value across the learner, rollout model, and reference model, indexed by k:

$$V_k = u_k(x_t, s, t) + (t-s)\,\mathrm{sg}[d], \qquad F_k = x_t - t\, V_k, \qquad k \in \{\theta, \mathrm{old}, \mathrm{ref}\}. \tag{6}$$

Here V_k is the corrected instantaneous-velocity representation and F_k its corresponding x_0-space prediction. As the same correction is used for all three networks, it cancels in every pairwise difference:

$$F_\theta - F_k = -t\,(u_\theta - u_k), \qquad V_\theta - V_k = u_\theta - u_k, \qquad k \in \{\mathrm{old}, \mathrm{ref}\}. \tag{7}$$

The rollout and reference anchors therefore compare the average-velocity networks directly on the sampled interval (s, t). With a separate derivative for each network, each anchor residual would instead contain an additional term $t(t-s)(d_\theta - d_k)$ depending on the sampled direction v_t; Section 4 shows why this matters.

3.2 Advantage-weighted least squares on the induced prediction

With the prediction equation 6, the MFA loss is the AdvantageFlow quadratic equation 2:

$$\ell_\theta = A\|F_\theta - x_0\|_2^2 + \gamma\|F_\theta - F_{\mathrm{old}}\|_2^2 + \lambda\|F_\theta - F_{\mathrm{ref}}\|_2^2, \qquad \mathcal{L}_{\mathrm{MFA}}(\theta) = \mathbb{E}_{x_0 \sim \pi_{\mathrm{old}}}[\ell_\theta]. \tag{8}$$

Within one update, the samples, the advantages, and the two anchor predictions are held fixed.

MFA is an advantage-weighted MeanFlow regression. Denote the regression target of MeanFlow training (Geng et al., 2025) by

$$\bar{u} = v_t - (t - s)\, d. \tag{9}$$

Since $x_t - x_0 = t\, v_t$, we have $F_\theta - x_0 = -t(u_\theta - \bar{u})$. Combined with equation 7, this turns the loss into an exact identity in average-velocity space:

$$\ell_\theta = t^2 \Big(A \, \|u_\theta - \bar{u}\|_2^2 + \gamma \, \|u_\theta - u_{\text{old}}\|_2^2 + \lambda \, \|u_\theta - u_{\text{ref}}\|_2^2 \Big). \tag{10}$$

The data term is the standard MeanFlow loss weighted by the advantage, and the anchors keep the deployed map close to the rollout and reference maps on the sampled interval. The prediction F is only a device to derive equation 10; what is trained and regularized is u_θ itself. When $s = t$, the correction vanishes and equation 10 reduces to AdvantageFlow on the boundary velocity $u_\theta(x_t, t, t)$ (Appendix A.1). The positive-length intervals are therefore what is new for few-step generators.

Per-sample solution and signed advantages. Treat $u = u_\theta(x_t, s, t)$ as a free vector. The loss equation 10 has curvature $2t^2(A + \gamma + \lambda)$, so whenever $A + \gamma + \lambda > 0$ it is strictly convex and has the unique minimizer

$$u^\star = \frac{A\, \bar{u} + \gamma\, u_{\text{old}} + \lambda\, u_{\text{ref}}}{A + \gamma + \lambda}. \tag{11}$$

The three weights sum to one. For $A > 0$, $u^\star$ moves toward the target $\bar{u}$. For $A < 0$, the weight on $\bar{u}$ is negative, so $u^\star$ is pushed away from the target, past the rollout output. A poor sample therefore produces a push away from itself instead of being ignored, and the loss remains convex. We use $\gamma = 1.1$, $\lambda = 10^{-3}$, and $A \in [-1, 1]$, so $A + \gamma + \lambda \geq 0.101$ always holds. Section 4 shows that γ also controls how far each update can move.

Adaptive scale. As in the DiffusionNFT trainer, the image implementation divides each sample's loss by a detached positive scale:

$$\mathcal{L}_{\text{train}} = \mathbb{E}[\ell_\theta / w], \qquad w = \max\big(\text{sg}[\text{mean}\,|F_\theta - x_0|], 10^{-5}\big), \tag{12}$$

where the mean is over latent coordinates. The three terms of a sample share the same w, so w does not change the per-sample minimizer equation 11. It only changes how much each sample contributes to the gradient. The ablations show that this matters for stability (Section 5.2). The analysis below is stated for the unscaled loss.

3.3 Samples, advantages, and intervals

We keep the outer loop of MeanFlowNFT (Huang et al., 2026b), so any difference from it comes from the objective alone.

Rollouts. Starting from Gaussian noise, the frozen rollout model runs the N-step sampler on a decreasing grid $1 = t_N > \cdots > t_0 = 0$:

$$x_{t_{i-1}} = x_{t_i} - (t_i - t_{i-1})\, u_{\text{old}}(x_{t_i}, t_{i-1}, t_i), \qquad i = N, \ldots, 1, \tag{13}$$

with $N = 4$ and no CFG. The rollout parameters are an EMA of the learner and are refreshed between updates. The reference is the frozen AnyFlow initialization (Gu et al., 2026). Fine-tuning uses LoRA (Hu et al., 2022).

Advantages. For L prompts with K images each, rewards are centered within each prompt and divided by a global standard deviation:

$$A^{i,k} = \text{clip}\Big(\frac{R^{i,k} - \widehat{R}^i}{\max(Z, \varepsilon_Z)}, -1, 1 \Big), \qquad \widehat{R}^i = \frac{1}{K} \sum_k R^{i,k}, \qquad Z^2 = \frac{1}{LK} \sum_{i,k} \big(R^{i,k} - \widehat{R}^i\big)^2. \tag{14}$$

With several rewards, each is standardized separately and the results are combined with fixed weights before clipping. The advantages are signed: an image worse than its prompt average receives $A < 0$.

Intervals. A few-step sampler takes jumps of positive length, so training must include such intervals. Each sample receives one of three interval types:

$$s = t \text{ (boundary)}, \qquad s = 0 \text{ (full jump to data)}, \qquad 0 < s < t \text{ (general interval)}, \tag{15}$$

with probabilities $(0.5, 0.25, 0.25)$ and the time shift of the base scheduler. Boundary samples keep the instantaneous velocity accurate. The other two types train the maps that the sampler actually runs. Algorithm 1 in Appendix B summarizes one update.

4 WHAT DOES MEANFLOWADVANTAGE LEARN?

The identity equation 10 holds exactly for every sample. We now take expectations and answer three questions. Which function does the loss drive the network toward? How large is each update, differing from MeanFlowNFT? When does the few-step sampler inherit the improvement? The analysis is at the population level, with a free function in place of the network. We write $Z = (x_t, s, t, c)$ for the network input and treat the advantage as $A = A(x_0, c)$. Proofs are in Appendix A.

Assumptions. We use three assumptions: *(A1) Exact correction:* $d = \partial_t u_{\text{old}} + (\partial_x u_{\text{old}}) v_t$, which a JVP computes exactly while equation 5 has error up to $O(\delta^2)$; *(A2) Positive weights:* $A + \gamma > 0$ for every sample; and *(A3) Calibrated rollout:* $\mathbb{E}_{\pi_{\text{old}}}[\bar{u} \mid Z] = u_{\text{old}}(Z)$. Assumption (A3) means that the rollout network is a MeanFlow fixed point on its own samples. Under (A1), this corresponds to the MeanFlow identity equation 4 for the velocity field of π_{old}. It is exact only for a perfectly calibrated rollout; we discuss this gap at the end of the section.

For each prompt c, define the reweighted distribution

$$q(x_0 \mid c) = \frac{\gamma + A(x_0, c)}{\gamma + \bar{A}_c}\, \pi_{\text{old}}(x_0 \mid c), \qquad \bar{A}_c = \mathbb{E}_{\pi_{\text{old}}}[A \mid c]. \tag{16}$$

q upweights positive-advantage samples and downweights negative ones; smaller γ strengthens the reweighting. Let $v_q(x_t, t, c) = \mathbb{E}_q[\epsilon - x_0 \mid x_t, t, c]$ be the rectified-flow velocity that transports noise to q.

Theorem 1 (MFA fits the MeanFlow target of a reward-reweighted distribution)**.** *Assume (A1)–(A3). There is a constant C that does not depend on θ such that*

$$\mathcal{L}_{\text{MFA}}(\theta)\big|_{\lambda=0} = \mathbb{E}_c\Big[(\gamma + \bar{A}_c)\, \mathbb{E}_q\big[t^2 \|u_\theta - \bar{u}\|_2^2 \,\big|\, c\big]\Big] + C. \tag{17}$$

For any $\lambda \geq 0$, the loss has a unique minimizer over free functions $u(Z)$:

$$u^+ = \alpha\, u_q + (1 - \alpha)\, u_{\text{ref}}, \qquad \alpha = \frac{\gamma + \mathbb{E}_{\pi_{\text{old}}}[A \mid Z]}{\gamma + \lambda + \mathbb{E}_{\pi_{\text{old}}}[A \mid Z]}, \tag{18}$$

where $u_q = v_q - (t - s)\big[\partial_t u_{\text{old}} + (\partial_x u_{\text{old}})\, v_q\big]$.

For $\lambda = 0$, MFA on rollout samples is equivalent to ordinary MeanFlow training under q, without sampling from q: u_q is the corresponding MeanFlow target with the rollout derivative stop-gradiented. Positive λ blends this target with the reference map. The shared correction is essential: because $u_\theta - u_{\text{old}}$ depends only on Z equation 7, the rollout anchor becomes a MeanFlow regression up to a constant; separate derivatives introduce dependence on v_t and break this equivalence.

Proposition 1 (Update direction and step size)**.** *Assume (A1)–(A3), and let $\sigma_A^2(Z) = \mathrm{Var}_{\pi_{\text{old}}}(A \mid Z)$ and $\sigma_{\bar{u}}^2(Z) = \operatorname{tr} \mathrm{Cov}_{\pi_{\text{old}}}(\bar{u} \mid Z)$. Then*

$$\begin{aligned} u^+ - u_{\text{old}} &= \frac{\mathrm{Cov}_{\pi_{\text{old}}}(A, \bar{u} \mid Z) + \lambda(u_{\text{ref}} - u_{\text{old}})}{\gamma + \lambda + \mathbb{E}_{\pi_{\text{old}}}[A \mid Z]}, \\ \|u^+ - u_{\text{old}}\|_2 &\leq \frac{\sigma_A \sigma_{\bar{u}} + \lambda \|u_{\text{ref}} - u_{\text{old}}\|_2}{\gamma + \lambda + \mathbb{E}_{\pi_{\text{old}}}[A \mid Z]}. \end{aligned} \tag{19}$$

The update direction is the covariance between the advantage and the MeanFlow target. Among all images that are consistent with the noisy input Z, the map moves toward the targets of high-advantage images and away from those of low-advantage images. The step size scales as $1/\gamma$, so γ acts as an inverse step size, or trust-region radius, on the deployed map. The step is also bounded by the spread of advantages at Z: if all images consistent with Z are equally good, nothing moves. A constant advantage $A \equiv 1$ has zero covariance, so for $\lambda = 0$ it gives $u^+ = u_{\text{old}}$. It carries no reward information, and the loss only refits the rollout samples.

Proposition 2 (Relation to MeanFlowNFT)**.** *Apply the NFT loss (Zheng et al., 2026) to the induced velocity equation 6 as in MeanFlowNFT, with parameter $\beta > 0$, optimality probability $r \in [0, 1]$, no reference term, and no adaptive weights. Under (A1) and (A3), its minimizer over free functions is*

$$u_{\text{NFT}}^+ = u_{\text{old}} + \frac{1}{\beta}\, \mathrm{Cov}_{\pi_{\text{old}}}(A, \bar{u} \mid Z), \qquad A = 2r - 1. \tag{20}$$

Compared with equation 19 at $\lambda = 0$, both objectives move in the same direction, but NFT uses the fixed gain $1/\beta$ while MFA uses $1/(\gamma + \mathbb{E}_{\pi_{\text{old}}}[A \mid Z])$. By Theorem 1, the MFA target at every input is the MeanFlow target of one fixed distribution q; the NFT target has this form only if $\mathbb{E}_{\pi_{\text{old}}}[A \mid Z]$ does not depend on Z. For example, with $A = 2r - 1$ and $\gamma = 1$, $q \propto r\,\pi_{\text{old}}$ is exactly the positive policy that motivates DiffusionNFT. MFA regresses onto its MeanFlow target u_{π^+} everywhere, whereas NFT regresses onto $u_{\text{old}} + (2\rho/\beta)(u_{\pi^+} - u_{\text{old}})$ with $\rho(Z) = \mathbb{E}_{\pi_{\text{old}}}[r \mid Z]$. That target falls short of u_{π^+} where $2\rho < \beta$ and overshoots it where $2\rho > \beta$ (Appendix A.4). This does not make MFA uniformly better, but its target has a simple distributional meaning at every noise level.

Theorem 2 (When the sampler inherits the improvement)**.** *Let $\lambda = 0$, and let u^* be a fixed point of the target in Theorem 1 when its own derivative is used:*

$$u^* = v_q - (t - s)\big[\partial_t u^* + (\partial_x u^*)\, v_q\big] \quad \textit{for all } 0 \le s < t \le 1. \tag{21}$$

Under the regularity conditions of Lemma 1, u^* is the exact average velocity of the flow of v_q. For every number of steps N and every grid, the sampler equation 13 with u^* therefore draws exact samples from q. For each prompt,

$$\mathbb{E}_q[R \mid c] - \mathbb{E}_{\pi_{\text{old}}}[R \mid c] = \frac{\text{Cov}_{\pi_{\text{old}}}(A, R \mid c)}{\gamma + \bar{A}_c}, \tag{22}$$

which is nonnegative whenever A is a nondecreasing function of a scalar reward R.

Condition equation 21 is the same stop-gradient fixed point that MeanFlow pretraining aims for: the target uses the current network's derivative, and a network that matches its own target is an exact flow map. Theorem 2 has two practical consequences. First, the condition must hold for *every* interval. Boundary training alone ($s = t$) only pins down v_q and leaves the maps used by the sampler unconstrained, which is why equation 15 includes positive-length intervals. Second, the target distribution q does not depend on N, so at the fixed point one update improves every sampling budget at once. Together with Proposition 1, it also shows the role of γ: a smaller γ gives a larger possible improvement equation 22, but also larger and less stable steps equation 19.

What the analysis does not cover. The theory assumes an exact correction (A1), a calibrated rollout (A3), and free functions, whereas training uses finite differences, LoRA, and the sample-dependent scale equation 12. With multiple rewards, equation 22 applies only to the combined score. We therefore test the main predictions in Section 5.2: the need for the correction and positive-length intervals, the absence of an update under constant advantages, and consistency across sampling budgets. Appendix A.6 gives an additional variance-reduction result.

5 Experiments

We evaluate MFA as on-policy RL on SD3.5-Medium (Sections 5.1–5.2). Appendix E applies the same objective to DNA promoters, both as teacher-free on-policy RL for a generator defined on a manifold and as reward-graded distillation.

5.1 Text-to-image generation with SD3.5-Medium

Setup. We start from the merged AnyFlow pretraining and on-policy distillation adapters for SD3.5-Medium (Esser et al., 2024; Gu et al., 2026) and train a fresh LoRA with four-step, CFG-free rollouts. Training uses 512×512 images, $L = 48$ prompt groups with $K = 24$ images each, and AdamW with learning rate 3×10^{-6}. PickScore, HPSv2, and CLIPScore are the training rewards, with equal weights (Kirstain et al., 2023; Wu et al., 2023; Hessel et al., 2021). ImageReward, Aesthetic Score, HPSv3, GenEval2, and OCR are evaluated but never optimized (Xu et al., 2023). DrawBench (Saharia et al., 2022) is used for online evaluation. The outer loop and hyperparameters match MeanFlowNFT, and Appendix C lists them all.

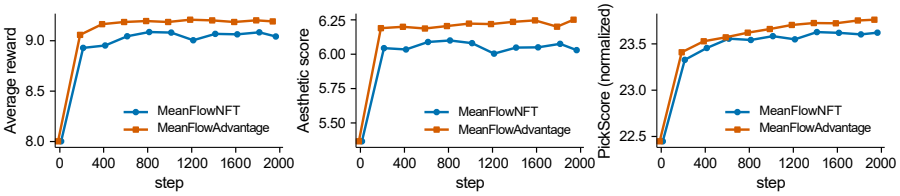
Main results. Table 1 compares MFA with the published baselines and with five matched evaluations (†). Under the matched evaluation, MFA improves on both MeanFlowNFT and TDM-R1 on all eight metrics. Relative to MeanFlowNFT, the largest gains are on Aesthetic Score ($+0.37$) and HPSv3 ($+0.26$), neither of which is a training reward, indicating that the improvement is not

Table 1: Text-to-image results on SD3.5-Medium. All rows without a mark are copied from Table 1 of MeanFlowNFT (Huang et al., 2026b), which evaluates at 1024×1024 following DiffusionNFT. †Evaluated with our pipeline (same prompts, resolution, seeds, and scorers as MFA). Among few-step models, **bold** marks the best value and underline the second best. PickScore is the raw logit.

Method	ImageReward↑	CLIPScore↑	Aesthetic↑	PickScore↑	HPSv2↑	HPSv3↑	GenEval2↑	OCR↑
Multi-step models (40 steps)								
SD3.5-M†	−0.463	0.240	5.178	20.77	0.208	2.769	0.104	0.134
+ DiffusionNFT† (Zheng et al., 2026)	1.426	0.299	5.492	23.55	0.330	13.893	0.224	0.639
Few-step models (4 steps)								
DMD (Yin et al., 2024)	0.924	0.284	5.506	22.28	0.287	11.716	0.204	0.400
CDM (Liu et al., 2026)	1.031	0.282	5.572	22.42	0.298	12.519	0.202	0.323
AnyFlow† (Gu et al., 2026)	1.113	0.289	5.420	22.48	0.297	12.069	0.190	0.452
RTDMD (Huang et al., 2026a)	1.232	0.278	6.129	23.28	0.327	13.925	0.204	0.297
Rdm (Fan et al., 2026)	0.724	0.272	5.754	22.07	0.276	11.212	0.162	0.376
DMD + DiffusionNFT	0.716	0.284	5.378	21.96	0.271	9.699	0.225	0.487
CDM + DiffusionNFT	0.146	0.275	4.834	21.38	0.214	−3.283	0.210	0.330
AnyFlow + DiffusionNFT	1.239	0.292	5.949	23.09	0.292	12.138	0.234	0.595
TDM-R1† (Luo et al., 2026)	1.444	0.293	5.942	22.80	0.324	13.512	0.219	0.623
MeanFlowNFT† (Huang et al., 2026b)	1.446	0.296	5.928	23.50	0.327	13.676	0.218	0.627
MeanFlowAdvantage (ours)	**1.451**	**0.297**	**6.300**	**23.63**	**0.330**	**13.933**	0.229	**0.631**



Figure 1: Qualitative comparison on prompts from GenEval, OCR, and DrawBench.



(a) Held-out Average Score (b) Held-out Aesthetic Score (c) Held-out PickScore

Figure 2: Held-out DrawBench scores during training, MeanFlowAdvantage versus MeanFlowNFT under the same outer loop.

confined to the optimized objectives. Among all few-step models, MFA ranks first on seven of the eight metrics, with particularly strong results on Aesthetic Score, PickScore, HPSv2, HPSv3, and OCR. With four NFEs, MFA exceeds the 40-step DiffusionNFT on ImageReward, CLIPScore, Aesthetic Score, HPSv2, and HPSv3 and is within 0.01 on PickScore, at one tenth of the sampling cost. DiffusionNFT remains better on GenEval2 and OCR, both of which its multi-reward setup trains on directly. Qualitatively, MFA more often preserves the requested object count, attribute binding, and text (Figure 1). Figure 2 further shows that these gains persist through most of training rather than arising from a single selected checkpoint.

Table 2: Any-step evaluation on DrawBench.

NFE	IR↑	CLIP↑	Aes↑	Pick↑	HPSv2↑	HPSv3↑	GenEval2↑	OCR↑
1	0.489	0.271	5.675	21.692	0.248	4.533	0.160	0.140
2	1.377	0.288	**6.359**	23.328	0.316	12.602	0.225	0.532
4	1.451	0.297	6.300	23.637	0.330	13.933	0.229	**0.631**
8	1.458	**0.298**	6.174	**23.713**	**0.333**	14.218	0.236	0.625
16	1.467	**0.298**	6.084	23.679	0.332	**14.255**	0.240	0.621
32	**1.492**	0.297	6.045	23.646	0.332	14.207	**0.250**	0.625

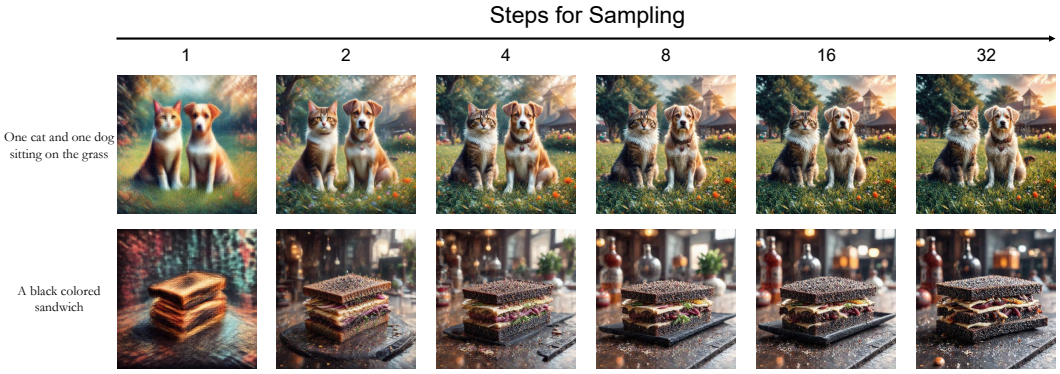


Figure 3: Qualitative test-time scaling of MeanFlowAdvantage across sampling budgets.

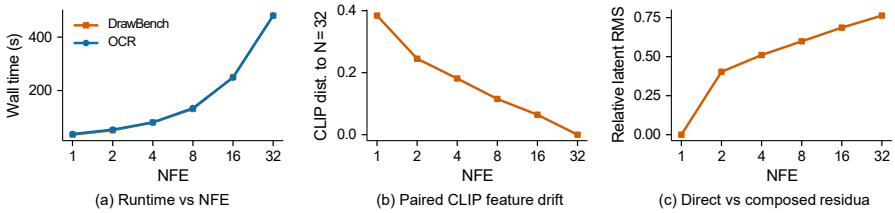


Figure 4: Any-step behavior of MFA: runtime, drift of samples across budgets, and semigroup residual of the learned flow map.

Any-step behavior. Theorem 2 predicts that, near its fixed point, one model should serve every sampling budget. Table 2 tests this with the checkpoint trained only with four-step rollouts. One step remains clearly harder. Two steps recover most of the quality, and from four to 32 steps the scores stay on a plateau: ImageReward, GenEval2, and HPSv3 keep improving slightly, while Aesthetic Score peaks at two steps. The model is therefore usable across budgets but is not an exact flow map. The semigroup probe in Figure 4 measures this directly. It compares the direct map $\Phi_{1\to 0}$ with compositions of N sub-steps (Appendix A.7); the relative residual grows from 0.40 at $N = 2$ to 0.76 at $N = 32$. The fixed point equation 21 is thus only approximately reached. Four steps give a good quality-cost trade-off. Figure 3 shows samples across budgets.

5.2 Testing the analysis on the image model

Each ablation changes one component of MFA under the same SD3.5-Medium initialization, fresh LoRA, 48 prompt groups per update, 2000 steps, and seed 42. Figure 5 shows the training curves, and Appendix D.1 reports the endpoints (Table S5). The results broadly follow the predictions of Section 4. **(i) The MeanFlow correction matters.** Removing the correction while keeping positive-length intervals lowers both the training reward (1.507 vs. 1.549) and held-out aggregate (8.89 vs. 8.98), while roughly doubling the gradient norm. Computing a separate derivative for each network reaches similar quality (9.04) but costs 23% more per step, suggesting that sharing is mainly an efficiency choice in practice, while its theoretical role is to preserve the equivalence in Theorem 1. **(ii) Positive-length intervals are necessary.** Boundary-only training constrains the instantaneous velocity but leaves the finite-interval maps used by the sampler unconstrained. The run collapses

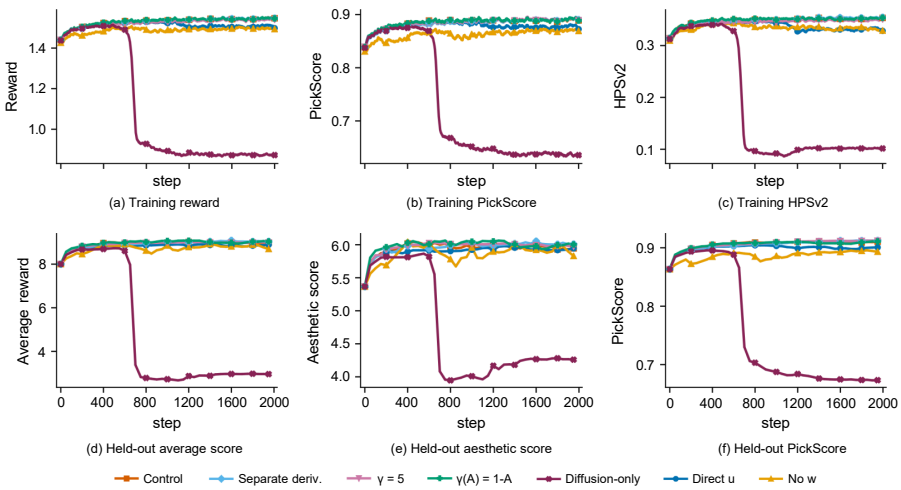


Figure 5: Training reward and held-out score of the matched MFA ablations. Boundary-only training collapses late; the positive-only and $A \equiv 1$ runs collapse within the first few dozen steps and are stopped at step 300.

after about 670 steps, ending with a held-out aggregate of 2.96 and a gradient norm above 400, consistent with Theorem 2. **(iii) Signed advantages provide the direction.** With $A \equiv 1$, Proposition 1 predicts no reward-directed update, and training collapses within 20 steps; keeping only positive advantages also collapses early (step 22), suggesting that negative samples matter beyond the population reweighting picture. In contrast, increasing γ from 1.1 to 5 or using $\gamma(A) = 1 - A$ gives endpoints close to the control, so the gain does not appear to come from a narrowly tuned trust region. **(iv) Adaptive scaling stabilizes optimization.** Removing w from equation 12 raises the final gradient norm from 0.97 to 11.97 and lowers the held-out aggregate to 8.70 without immediate collapse. Since w leaves the per-sample minimizer unchanged, its effect is on optimization stability rather than on the target itself.

5.3 Reward-filtered advantage-weighted promoter fine-tuning

We further test MFA on 1024-bp FANTOM5 promoters with a Riemannian MeanFlow generator and negative Sei profile MSE as reward. In teacher-free on-policy RL, MFA lowers the pretrained Sei MSE from 0.0714 to 0.0482 over three seeds and is statistically indistinguishable from MeanFlowNFT on the optimized reward, consistent with Proposition 2. For distillation, a frozen 10-step Sei-guided teacher provides improved endpoints; we keep pairs with $\Delta r \geq 0.002$ and use $A = \mathrm{clip}(10\Delta r, 0, 1)$. On identical pairs and advantages, MFA reaches 0.0467 Sei MSE versus 0.0570 for MeanFlowNFT, an 18% reduction. Full configurations and ablations are in Appendix E.

6 Conclusion

We introduced MeanFlowAdvantage, a signed advantage-weighted objective that aligns reward fine-tuning with the average-velocity representation deployed by MeanFlow. A single detached correction shared by the learner, rollout, and reference networks turns the prediction-space quadratic into an exact advantage-weighted MeanFlow regression, while keeping the rollout and reference anchors directly on the deployed finite-interval maps. At the population level, this objective is equivalent to fitting the MeanFlow target of a reward-reweighted distribution; its update direction is governed by an advantage–target covariance, and self-consistent fixed points define exact flow maps that transfer across sampling budgets. Empirically, MFA improves all eight reported metrics over matched MeanFlowNFT on SD3.5-Medium and matches or exceeds the 40-step DiffusionNFT baseline on six of eight with only four NFEs. The same formulation also extends beyond images, supporting both teacher-free RL and reward-graded distillation for DNA promoter generation. Open questions remain around approximate flow-map consistency, principled negative advantages for distillation, and the three-seed sequence-fidelity trend.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing, language refinement, LaTeX formatting, and mathematical consistency checking. They were not used as a substitute for experimental validation or independent verification of the reported results. The authors reviewed and verified all AI-assisted text, derivations, implementation details, references, and experimental claims, and take full responsibility for the final content of the paper.

ETHICS STATEMENT

The models inherit biases and failure modes of their pretrained generators and reward models. The promoter experiments evaluate public human genomic sequences computationally; Sei scores are not experimental validation of regulatory function.

REPRODUCIBILITY STATEMENT

The main algorithm and optimization objective are specified in Algorithm 1 and Section 3. Experimental configurations and implementation details are provided in Appendix C and Appendix B, complete proofs and theoretical assumptions are given in Appendix A, and additional image-model ablations and promoter evaluations are reported in Appendices D.1 and E.1. We will publish anonymized code, configuration files, evaluation scripts, and checkpoint information on github.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

Kathleen M. Chen, Aaron K. Wong, Olga G. Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature Genetics*, 54:940–949, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

Linqian Fan, Peiqin Sun, Tiancheng Wen, Shun Lu, and Chengru Song. Rdm: Re-conceptualizing distribution matching as a reward for diffusion distillation. *arXiv preprint arXiv:2603.28460*, 2026.

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 *Advances in Neural Information Processing Systems*, 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. AnyFlow: Any-step video diffusion model with on-policy flow map distillation. *arXiv preprint arXiv:2605.13724*, 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of EMNLP*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Yushi Huang, Xiangxin Zhou, Ruoyu Wang, Chi Zhang, Jun Zhang, and Tianyu Pang. Reinforcing few-step generators via reward-tilted distribution matching. *arXiv preprint arXiv:2605.26108*, 2026a.

Yushi Huang, Xiangxin Zhou, Jun Zhang, Liefeng Bo, and Tianyu Pang. MeanFlowNFT: Bringing forward-process RL to average-velocity generators. *arXiv preprint arXiv:2607.15273*, 2026b.

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.

Branislav Kveton, Anup Rao, Subhojyoti Mukherjee, Krishna Kumar Singh, and Viet Dac Lai. AdvantageFlow: Advantage-weighted least squares for RL in flow models. *arXiv preprint arXiv:2605.26013*, 2026.

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

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. In *Advances in Neural Information Processing Systems*, 2025.

Tao Liu, Hao Yan, Mengting Chen, Taihang Hu, Zhengrong Yue, Zihao Pan, Jinsong Lan, Xiaoyong Zhu, Ming-Ming Cheng, Bo Zheng, and Yaxing Wang. Continuous-time distribution matching for few-step diffusion distillation. *arXiv preprint arXiv:2605.06376*, 2026.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Yihong Luo, Tianyang Hu, Weijian Luo, and Jing Tang. TDM-R1: Reinforcing few-step diffusion models with non-differentiable reward. In *International Conference on Machine Learning*, 2026.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 2022.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46495–46513. PMLR, 2024.

Dongyeop Woo, Marta Skreta, Seonghyun Park, Sungsoo Ahn, and Kirill Neklyudov. Riemannian MeanFlow. *arXiv preprint arXiv:2602.07744*, 2026.

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

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusion NFT: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, 2026.

A PROOFS FOR SECTION 4

Setting. Throughout, c is drawn from the prompt distribution, $x_0 \sim \pi_{\text{old}}(\cdot \mid c)$, $\epsilon \sim \mathcal{N}(0, I)$, and (s, t) is drawn independently of (x_0, ϵ). We write $x_t = (1-t)x_0 + t\epsilon$, $v_t = \epsilon - x_0$, and $Z = (x_t, s, t, c)$. The advantage is a function $A = A(x_0, c)$. Under (A1), $d = D(Z) + J(Z)\, v_t$ with $D = \partial_t u_{\text{old}}(Z)$ and $J = \partial_x u_{\text{old}}(Z)$; both are functions of Z only. All second moments are assumed finite. Conditional expectations under π_{old} are written $\mathbb{E}[\cdot \mid Z]$.

A.1 AVERAGE-VELOCITY FORM OF THE LOSS AND BOUNDARY REDUCTION

By equation 6, $F_\theta = x_t - t u_\theta - t(t-s)d$. Since $x_t - x_0 = t\, v_t$,

$$F_\theta - x_0 = t\, v_t - t\, u_\theta - t(t-s)d = -t\big(u_\theta - \bar{u}\big), \qquad \bar{u} = v_t - (t-s)d. \tag{23}$$

The anchor residuals follow from equation 6 because the correction terms cancel: $F_\theta - F_k = -t(u_\theta - u_k)$. Substituting both into equation 8 gives equation 10. This identity holds for any value of d, including the finite difference. At $s = t$, the factor $t - s$ vanishes, so $\bar{u} = v_t$ and $F_k = x_t - t u_k(x_t, t, t)$ for every network. The loss equation 10 then equals the AdvantageFlow loss equation 2 applied to the boundary predictor $x_t - t u_\theta(x_t, t, t)$, for the same samples, advantages, and coefficients. The scaled loss equation 12 reduces in the same way when the comparison uses the same denominator.

A.2 PROOF OF THEOREM 1

Step 1: the rollout anchor is a MeanFlow regression. Let $g(Z)$ be any function of Z. Expanding the square and using (A3),

$$\mathbb{E}\big[\|g - \bar{u}\|_2^2 \mid Z\big] = \|g - u_{\text{old}}\|_2^2 + \mathbb{E}\big[\|u_{\text{old}} - \bar{u}\|_2^2 \mid Z\big] + 2\big\langle g - u_{\text{old}},\, u_{\text{old}} - \mathbb{E}[\bar{u} \mid Z]\big\rangle, \tag{24}$$

and the last term is zero. By equation 7, the learner output $u_\theta(Z)$ is such a function, so

$$t^2\|u_\theta - u_{\text{old}}\|_2^2 = \mathbb{E}\big[t^2\|u_\theta - \bar{u}\|_2^2 \mid Z\big] - \mathbb{E}\big[t^2\|u_{\text{old}} - \bar{u}\|_2^2 \mid Z\big]. \tag{25}$$

Taking expectations in equation 10 with $\lambda = 0$ gives

$$\mathcal{L}_{\text{MFA}}(\theta) = \mathbb{E}\big[(A+\gamma)\, t^2\|u_\theta - \bar{u}\|_2^2\big] + C, \qquad C = -\gamma\, \mathbb{E}\big[t^2\|u_{\text{old}} - \bar{u}\|_2^2\big]. \tag{26}$$

Step 2: change of measure. Let $h(x_0, \epsilon, s, t, c) = t^2\|u_\theta(Z) - \bar{u}\|_2^2$. Because A depends only on (x_0, c) and (ϵ, s, t) are independent of x_0,

$$\mathbb{E}\big[(A+\gamma)h\big] = \mathbb{E}_c\Big[\int (\gamma + A)\, \pi_{\text{old}}(x_0 \mid c)\, \mathbb{E}_{\epsilon,s,t}[h]\, \mathrm{d}x_0\Big] = \mathbb{E}_c\Big[(\gamma + \bar{A}_c)\, \mathbb{E}_q\big[h \mid c\big]\Big], \tag{27}$$

where q is defined in equation 16. By (A2), q is a valid probability distribution. This proves equation 17.

Step 3: minimizer. For $\lambda \geq 0$, the conditional loss given Z is

$$t^2\Big(\mathbb{E}\big[A\|u - \bar{u}\|_2^2 \mid Z\big] + \gamma\|u - u_{\text{old}}\|_2^2 + \lambda\|u - u_{\text{ref}}\|_2^2\Big), \tag{28}$$

a quadratic in u with Hessian $2t^2(\mathbb{E}[A \mid Z] + \gamma + \lambda)I$. This is positive definite for $t > 0$ by (A2). Its unique minimizer is

$$u^+ = \frac{\mathbb{E}[A\bar{u} \mid Z] + \gamma u_{\text{old}} + \lambda u_{\text{ref}}}{\mathbb{E}[A \mid Z] + \gamma + \lambda}. \tag{29}$$

By (A3), $\mathbb{E}[A\bar{u} \mid Z] + \gamma u_{\text{old}} = \mathbb{E}[(A+\gamma)\bar{u} \mid Z]$. Under q, the joint density of (x_0, ϵ, s, t) given c is $(\gamma + A)/(\gamma + \bar{A}_c)$ times that under π_{old}. By Bayes' rule, the conditional law given Z is reweighted by $\gamma + A$; the normalizer $\gamma + \bar{A}_c$ depends only on c, which is part of Z. Hence

$$\mathbb{E}[(A+\gamma)\bar{u} \mid Z] = \mathbb{E}[A + \gamma \mid Z]\, \mathbb{E}_q[\bar{u} \mid Z]. \tag{30}$$

Since D and J are functions of Z, $\mathbb{E}_q[\bar{u} \mid Z] = \mathbb{E}_q[v_t \mid Z] - (t-s)\big(D + J\, \mathbb{E}_q[v_t \mid Z]\big)$. Because (s, t) is independent of (x_0, ϵ), $\mathbb{E}_q[v_t \mid Z] = v_q(x_t, t, c)$, so $\mathbb{E}_q[\bar{u} \mid Z] = u_q$. Substituting into equation 29 gives $u^+ = \alpha u_q + (1-\alpha)u_{\text{ref}}$ with the stated α. □

Why the correction must be shared. Suppose the learner and the rollout used separate corrections d_θ and d_{old}. The rollout residual would be $u_\theta - u_{\text{old}} + (t-s)(d_\theta - d_{\text{old}})$, and under (A1) $d_\theta - d_{\text{old}} = \partial_t(u_\theta - u_{\text{old}}) + \partial_x(u_\theta - u_{\text{old}})\, v_t$. This residual depends on v_t and not only on Z, so Step 1 fails: the cross term no longer vanishes, and the anchor is no longer a MeanFlow regression on rollout samples.

A.3 Proof of Proposition 1

Subtracting u_{old} from equation 29,

$$u^+ - u_{\text{old}} = \frac{\mathbb{E}[A\bar{u} \mid Z] - \mathbb{E}[A \mid Z]\, u_{\text{old}} + \lambda(u_{\text{ref}} - u_{\text{old}})}{\mathbb{E}[A \mid Z] + \gamma + \lambda}. \tag{31}$$

By (A3), $\mathbb{E}[A\bar{u} \mid Z] - \mathbb{E}[A \mid Z]\,\mathbb{E}[\bar{u} \mid Z] = \mathrm{Cov}(A, \bar{u} \mid Z)$, which gives the identity in equation 19. For the bound, take any unit vector e. By Cauchy–Schwarz, $e^\top \mathrm{Cov}(A, \bar{u} \mid Z) = \mathrm{Cov}(A, e^\top \bar{u} \mid Z) \le \sigma_A \sqrt{\mathrm{Var}(e^\top \bar{u} \mid Z)} \le \sigma_A \sigma_{\bar{u}}$. Taking the supremum over e bounds the norm of the covariance vector. The triangle inequality and the positivity of the denominator (A2) complete the proof. □

A.4 Proof of Proposition 2 and the relation to prediction-space NFT

MeanFlowNFT forms the implicit positive and negative velocities $V^\pm = V_{\text{old}} \pm \beta(V_\theta - V_{\text{old}})$ and minimizes $r\|V^+ - v_t\|_2^2 + (1-r)\|V^- - v_t\|_2^2$. With the shared correction, $V_\theta - V_{\text{old}} = u_\theta - u_{\text{old}} =: \Delta$, a function of Z, and $V_{\text{old}} - v_t = u_{\text{old}} - \bar{u} =: e$. Hence

$$r\|\beta\Delta + e\|_2^2 + (1-r)\|-\beta\Delta + e\|_2^2 = \beta^2\|\Delta\|_2^2 + 2\beta A\langle\Delta, e\rangle + \|e\|_2^2, \qquad A = 2r - 1. \tag{32}$$

Conditioning on Z and minimizing over Δ gives $\Delta = -\mathbb{E}[Ae \mid Z]/\beta = \big(\mathbb{E}[A\bar{u} \mid Z] - \mathbb{E}[A \mid Z]u_{\text{old}}\big)/\beta$. By (A3), this equals $\mathrm{Cov}(A, \bar{u} \mid Z)/\beta$, which proves equation 20. □

The positive-policy example. Let $A = 2r - 1$ and $\gamma = 1$. Then $\gamma + A = 2r$ and $q \propto r\,\pi_{\text{old}} = \pi^+$. Write $\rho = \mathbb{E}[r \mid Z]$ and $u_{\pi^+} = \mathbb{E}_{\pi^+}[\bar{u} \mid Z]$. By (A3), $\mathrm{Cov}(A, \bar{u} \mid Z) = 2\big(\mathbb{E}[r\bar{u} \mid Z] - \rho\, u_{\text{old}}\big) = 2\rho\,(u_{\pi^+} - u_{\text{old}})$. The NFT minimizer is therefore $u_{\text{old}} + (2\rho/\beta)(u_{\pi^+} - u_{\text{old}})$. It equals u_{π^+} only where $2\rho(Z) = \beta$. By Theorem 1, the MFA minimizer is u_{π^+} at every Z.

Prediction-space form. Multiplying by t^2 and dropping θ-independent terms, equation 32 is equivalent to $\beta A\|F_\theta - x_0\|_2^2 + \beta(\beta - A)\|F_\theta - F_{\text{old}}\|_2^2$. After dividing by β, NFT is the MFA quadratic with the sample-dependent anchor weight $\gamma(A) = \beta - A$ and constant total curvature β. At $\beta = 1$, it matches the variant $\gamma(A) = 1 - A$ with $\lambda = 0$. Separate adaptive denominators for the positive and negative branches break this equivalence. The same expansion, with the data target y_κ, anchor b, and reference weight k, gives the NFT optimum in equation 39.

A.5 Flow-map consistency and proof of Theorem 2

For a velocity field v, define $\mathcal{I}_v[u] = u + (t-s)\big(\partial_t u + (\partial_x u)v\big)$.

Lemma 1 (Flow-map consistency)**.** *Suppose v admits a unique flow on $[0,1]$, and u is continuously differentiable for $s < t$ and bounded as $t \downarrow s$. If $\mathcal{I}_v[u] = v$ along every trajectory, then $u(x_t, s, t) = (x_t - x_s)/(t-s)$, where x_τ is the trajectory of v through x_t.*

Proof of the lemma. Fix s and let $U(\tau) = (\tau - s)\,u(x_\tau, s, \tau)$ for $\tau > s$. By the chain rule and the hypothesis, $\mathrm{d}U/\mathrm{d}\tau = \mathcal{I}_v[u](x_\tau, s, \tau) = v(x_\tau, \tau)$. Boundedness of u gives $U(\tau) \to 0$ as $\tau \downarrow s$. Integrating from s to t yields $(t-s)\,u(x_t, s, t) = \int_s^t v(x_\tau, \tau)\,\mathrm{d}\tau = x_t - x_s$. □

Proof of Theorem 2. Condition equation 21 says $\mathcal{I}_{v_q}[u^*] = v_q$. By Lemma 1, $x_t - (t-s)u^*(x_t, s, t) = x_s$ on every trajectory of v_q, so each step of equation 13 with u^* is the exact flow map of v_q over its interval. A composition over any grid is therefore the exact flow from $t = 1$ to $t = 0$. The rectified-flow velocity v_q transports $\mathcal{N}(0, I)$ at $t = 1$ to q at $t = 0$, so the sampler draws from q for every N. For the reward,

$$\mathbb{E}_q[R \mid c] - \mathbb{E}_{\pi_{\text{old}}}[R \mid c] = \frac{\mathbb{E}_{\pi_{\text{old}}}[(\gamma + A)R \mid c]}{\gamma + \bar{A}_c} - \mathbb{E}_{\pi_{\text{old}}}[R \mid c] = \frac{\mathrm{Cov}_{\pi_{\text{old}}}(A, R \mid c)}{\gamma + \bar{A}_c}. \tag{33}$$

If $A = \phi(R)$ with ϕ nondecreasing, Chebyshev's association inequality gives $\mathrm{Cov}(\phi(R), R) \ge 0$. The clipped, prompt-centered advantage equation 14 is of this form for a scalar reward when the prompt mean and the global scale are treated as fixed. □

A.6 VARIANCE REDUCTION OF THE SHARED ANCHOR

Proposition 3 (Shared-anchor variance reduction)**.** *Freeze the rollout model and the shared correction, and let $J_\theta = -t\,\partial_\theta u_\theta(Z)$. If $\mathbb{E}[F_{\mathrm{old}} - x_0 \mid Z] = 0$, then*

$$\mathbb{E}\|F_\theta - x_0\|_2^2 = \mathbb{E}\|F_\theta - F_{\mathrm{old}}\|_2^2 + C \tag{34}$$

for a θ-independent constant C. Moreover, the anchor gradient $2J_\theta^\top(F_\theta - F_{\mathrm{old}})$ is the conditional expectation, given Z, of the regression gradient $2J_\theta^\top(F_\theta - x_0)$, so its covariance is no larger in the positive-semidefinite order.

Proof. The condition is (A3) written in prediction space, since $F_{\mathrm{old}} - x_0 = -t(u_{\mathrm{old}} - \bar{u})$. Let $e = F_{\mathrm{old}} - x_0$. By equation 7, $F_\theta - F_{\mathrm{old}}$ is a function of Z, so $\mathbb{E}\langle F_\theta - F_{\mathrm{old}}, e\rangle = \mathbb{E}\langle F_\theta - F_{\mathrm{old}}, \mathbb{E}[e \mid Z]\rangle = 0$. Expanding $\|F_\theta - F_{\mathrm{old}} + e\|_2^2$ gives equation 34 with $C = \mathbb{E}\|e\|_2^2$. For the gradients, let $G = 2J_\theta^\top(F_\theta - x_0)$ and $\widetilde{G} = 2J_\theta^\top(F_\theta - F_{\mathrm{old}})$. Both J_θ and $F_\theta - F_{\mathrm{old}}$ are functions of Z, so $\mathbb{E}[G \mid Z] = \widetilde{G}$. The law of total covariance gives $\mathrm{Cov}(G) = \mathrm{Cov}(\widetilde{G}) + \mathbb{E}[\mathrm{Cov}(G \mid Z)] \succeq \mathrm{Cov}(\widetilde{G})$. □

This statement concerns the unscaled rollout anchor only. It does not imply lower variance for the full signed-advantage update with adaptive scaling.

A.7 SEMIGROUP RESIDUAL

For $\Phi^u_{s,t}(x) = x - (t-s)u(x,s,t)$, an exact flow map satisfies

$$\Phi^u_{r,t}(x) = \Phi^u_{r,s}\big(\Phi^u_{s,t}(x)\big), \qquad r < s < t. \tag{35}$$

By Lemma 1, a map that satisfies the fixed point equation 21 also satisfies equation 35. The residual of equation 35 therefore measures how far a trained model is from that fixed point. Figure 4 reports the relative latent RMS distance between the direct map $\Phi_{1\to 0}$ and uniform N-step compositions, which is zero by definition at $N = 1$. A small residual is not by itself evidence of reward improvement.

B IMPLEMENTATION NOTES

Algorithm 1 One MeanFlowAdvantage update (image model)

Require: learner θ, frozen rollout θ_{old} and reference θ_{ref}; rewards; γ, λ; group size K; steps N

1: Sample prompts; generate K images per prompt with equation 13
2: Score the images; compute clipped signed advantages with equation 14
3: **for** each detached (x_0, A, c) **do**
4: Draw (s,t) by equation 15 and ϵ; set $x_t = (1-t)x_0 + t\epsilon$
5: If $s = t$, set $d = 0$; otherwise compute d once from u_{old} by equation 5
6: Form $V_k = u_k + (t-s)d$ and $F_k = x_t - tV_k$ for $k \in \{\theta, \mathrm{old}, \mathrm{ref}\}$
7: Compute ℓ_θ / w with equation 8 and equation 12
8: **end for**
9: Update θ on the average loss
10: Refresh the rollout model: $\theta_{\mathrm{old}} \leftarrow \rho\,\theta_{\mathrm{old}} + (1-\rho)\,\theta$

Scheduler units and finite differences. With $T = 1000$, $t_{\mathrm{raw}} = Tt$ and $s_{\mathrm{raw}} = Ts$, the normalized and raw-time expressions agree:

$$x_t = (1-t)x_0 + t\epsilon, \qquad F = x_t - tV, \qquad (t_{\mathrm{raw}} - s_{\mathrm{raw}})d_{\mathrm{raw}} = (t-s)D_t u. \tag{36}$$

The reported finite-difference increment of 5 is in raw scheduler units, corresponding to $\delta = 0.005$. Spatial displacements must therefore use δv_t, not $5v_t$. The stencil equation 5 assumes $s \le t - \delta$ and $t + \delta \le 1$. Near a boundary, the stencil must be shortened or replaced by a valid one-sided difference; independently clamping its endpoints while retaining 2δ in the denominator is not the same estimator. On $s = t$, the correction is zero and the stencil is skipped.

Advantage and loss scaling. The final multi-reward advantage is clipped after aggregation. For constant γ, a lower bound $A \geq -\gamma - \lambda + \varepsilon_A$ enforces positive curvature; the reported $\gamma = 1.1$, $\lambda = 10^{-3}$ and $A \in [-1, 1]$ already satisfy it. For $\gamma(A) = 1 - A$, the curvature is $1 + \lambda$ and no additional floor is necessary. The shared denominator in equation 12 is evaluated in float64. It preserves the relative weights of the three terms within a sample, but changes the relative weights of different samples. It neither equalizes the three residual magnitudes nor preserves the unweighted population regression problem in general.

Ablations to isolate the construction. The direct-output control uses $F^{\text{direct}} = x_t - tu_\theta$, retaining the same clean target, regularizers, and time distribution. The diffusion-only control sets $\rho_{\text{diff}} = 1$. Prediction-space and velocity-space losses must be compared with their effective time weights reported: without other changes they differ by t^2. The completed matched endpoints for these controls are reported in Appendix Table S5.

C EXPERIMENTAL DETAILS

Tables S1, S2 and S4 collect the reported configurations, and Table S3 the per-seed DNA RL results. The image baseline follows the published Stage-3 protocol. Both methods use $K = 24$ samples per prompt; the corresponding EMA schedule, evaluation resolution, prompt lists, and checkpoint identifiers are fixed according to the reported Stage-3 protocol.

Table S1: SD3.5 Medium RL (Stage 3 protocol of MeanFlowNFT). Sampling and reward choices follow the baseline; the objectives and regularization coefficients differ.

	MeanFlowNFT	MeanFlowAdvantage
Init	AnyFlow + fresh LoRA	AnyFlow + fresh LoRA
LoRA rank / α	32 / 64	32 / 64
Image size (train)	512	512
Rollout NFE / CFG	4 / none	4 / none
Prompt groups L per update	48	48
Optimizer	AdamW	AdamW
Learning rate	3×10^{-6}	3×10^{-6}
(s, t) mix $(\rho_{\text{diff}}, \rho_{\text{cons}})$	$(0.5, 0.25)$	$(0.5, 0.25)$
Central-difference increment (raw units)	5	5
Shared D_t	yes	yes
Train rewards	Pick / HPS / CLIP	Pick / HPS / CLIP
β (NFT)	0.1	—
γ / λ (LS)	—	1.1 / 10^{-3}
Reference coefficient (objective-specific)	10^{-4}	10^{-3}
Advantage clip	$[-1, 1]$	$[-1, 1]$

Table S2: FANTOM5 on-policy DNA RL (no teacher). Both methods share every row above the line; only the loss coefficients differ.

	MeanFlowNFT	MeanFlowAdvantage
Init	RMF `ckpts/0120`	RMF `ckpts/0120`
Signals per update	8	8
Group size K	8	8
Rollout NFE	1	1
(s,t) mix (boundary, $s{=}1$, general)	$(0.5, 0.25, 0.25)$	$(0.5, 0.25, 0.25)$
t sampler	logit-normal$(-0.4, 1)$	logit-normal$(-0.4, 1)$
Noise draws per sample	2	2
Shared correction (JVP)	yes	yes
Tangent clip	100	100
Adaptive scale exponent p	0.5	0.5
Learning rate	10^{-4}	10^{-4}
Grad clip	0.5	0.5
Rollout EMA ρ	0.9	0.9
Advantage clip	$[-1, 1]$	$[-1, 1]$
Train steps (eval interval)	600 (25)	600 (25)
Seeds	$123, 7, 42$	$123, 7, 42$
β / reference weight	1.0 / 0.01	—
γ / λ	—	1.1 / 0.01

Table S3: Per-seed on-policy DNA results behind Table S6. Checkpoints are selected on valid ($n{=}256$) and reported on test ($n{=}512$), both with eval seed 7. Δ is the paired NFT$-$MFA test Sei MSE with a 95% sequence bootstrap interval; positive favors MFA.

	MeanFlowNFT		MeanFlowAdvantage		
Seed	Sei MSE↓	6-mer↑	Sei MSE↓	6-mer↑	Δ [95% CI]
123	0.0445	0.927	**0.0437**	**0.952**	$0.0008\ [-0.0031, 0.0046]$
7	**0.0526**	0.940	0.0537	**0.954**	$-0.0012\ [-0.0051, 0.0026]$
42	**0.0441**	0.944	0.0472	**0.953**	$-0.0031\ [-0.0069, 0.0005]$
Mean	**0.0471**	0.937	0.0482	**0.953**	—
Std.	0.0048	0.009	0.0051	**0.001**	—

All three Sei MSE intervals contain zero, so the two objectives are statistically indistinguishable on the training reward, as Proposition 2 predicts. The 6-mer ordering is consistent across seeds but is a three-seed trend, not a tested claim.

Table S4: FANTOM5 reward-filtered promoter distillation. Shared rows are identical for both losses; method columns list only the knobs that differ.

	MeanFlowNFT	MeanFlowAdvantage
Init	RMF `ckpts/0120`	RMF `ckpts/0120`
Sequence length	1024 bp	1024 bp
Train / valid / test chr.	not 8–10 / 10 / 8–9	not 8–10 / 10 / 8–9
Batch size	8	8
Student NFE	1 (unguided)	1 (unguided)
Teacher NFE / guidance	10 / 10	10 / 10
Teacher setting	x_1 look-ahead	x_1 look-ahead
Accept if Δr_{Sei}	≥ 0.002	≥ 0.002
Query time t	0	0
Advantage	$\mathrm{clip}(10\Delta r, 0, 1)$	$\mathrm{clip}(10\Delta r, 0, 1)$
Rollout EMA coefficient	1 (frozen)	1 (frozen)
Seed / eval seed	123 / 7	123 / 7
Learning rate	2.5×10^{-5}	2.5×10^{-5}
Grad clip	0.5	0.5
β	0.8	—
γ / λ	—	0.01 / 0.01
Target extrapolation	1.5	$\kappa = 1.5$
Max steps (selection)	100 (90)	100 (90)

Both DNA losses act on continuous RMF x_1 endpoint coordinates, using the convention of noise at $t = 0$ and data at $t = 1$. This differs from the image convention in equation 1. The learner calls the RMF endpoint predictor at $(t, s) = (0, 1)$; its data target is the guided endpoint, optionally extrapolated from the frozen rollout endpoint as specified in equation 38. Thus this adaptation does not use $F = x_t - tV$ at image time $t = 0$, which would have zero residual.

Figure S3b–c evaluates both methods every 10 steps through step 100. Under the EMA convention of Algorithm 1, $\rho = 1$ freezes the rollout parameters, while the student remains trainable. Evaluation samples from the updated learner; the frozen rollout model is used only to construct training pairs and anchor predictions.

D Additional text-to-image results

D.1 Matched image-model ablations

The complete matched ablation endpoints used by the discussion in Section 5.2 are reported below. These experiments provide robust directional insights that fully reinforce our primary findings.

For completed runs, training statistics are averages over the final 50 optimization steps and the held-out score is the final fixed-interval online evaluation (step 1950). The two unstable positive-only controls were stopped at step 300 after the predefined collapse screen; their final online evaluation is at step 250.

Figure S2 reports the optimization diagnostics for these matched runs. The secondary comparisons below complement the three core findings in Section 5.2.

Sharing the derivative is the efficient default. Computing a separate derivative correction for learner, rollout, and reference models reaches essentially the same reward regime as the shared correction, but increases wall-clock time per step from 73.3 to 89.9 seconds. Its slightly higher single-seed held-out aggregate is too small to establish a quality advantage. This matches the algebra in equation 7: sharing the correction makes the anchor residuals exact differences of the deployed average-velocity networks while avoiding two additional derivative estimates.

The anchor coefficient is not the main source of the gain. Increasing the fixed anchor coefficient from $\gamma = 1.1$ to $\gamma = 5$ leaves the final reward and held-out aggregate nearly unchanged, while the dynamic $\gamma(A) = 1 - A$ variant is also stable and reaches a comparable endpoint. These single-seed differences are too small to rank the stable variants, but they indicate that the principal effect is not

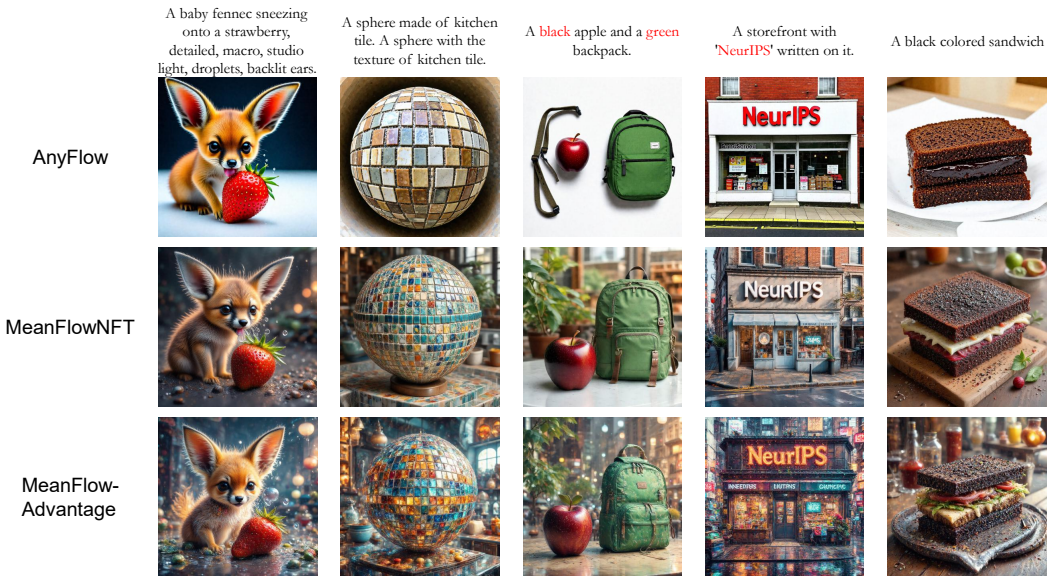


Figure S1: More Qualitative Comparison Cases. The prompts are taken from GenEval, OCR and DrawBench respectively,where we compare the corresponding MeanFlowNFT model with our model.

Table S5: Matched image-model ablations. "Reward" is the mean training reward over the final 50 recorded steps of each run; "Held-out Sum" is the fixed DrawBench online aggregate used for the ablation study. The early-stopped positive-only runs are not equal-budget endpoints.

Variant	Train step	Reward↑	Held-out Sum↑	Grad. norm↓	s/step↓
Shared induced V (control)	2000	1.5493	8.9841	0.97	73.3
Direct u (no induced correction)	2000	1.5071	8.8915	2.10	64.8
$s = t$ diffusion-only	2000	0.8734	2.9590	445.79	65.4
Separate derivative	2000	1.5486	9.0382	1.00	89.9
No adaptive w	2000	1.4959	8.6952	11.97	87.4
$\gamma = 5$	2000	1.5424	8.9936	0.84	72.9
Positive A only	300†	0.7956	2.3275	1.30	73.4
$A \equiv 1$	300†	0.8260	3.7154	24.17	73.6
$\gamma(A) = 1 - A$	2000	1.5482	9.0378	0.93	73.1

†Early-stopped after collapse screening; held-out evaluation is from step 250. The ablation aggregate uses the online normalized scoring convention and is not on the same numerical scale as the raw PickScore column in Table 1.

a narrowly tuned value of γ. We therefore retain $\gamma = 1.1$ as the conservative default and view $\gamma(A) = 1 - A$ as a useful follow-up rather than a claimed improvement.

Positive local curvature is necessary but not sufficient. All stable configurations maintain $A+\gamma+\lambda > 0$, as required by the pointwise quadratic analysis in Section 3.2. However, the $s = t$ diffusion-only run also retains a positive minimum coefficient while its data, reference, and gradient terms diverge late in training. The ablation therefore clarifies the scope of the theory: positive curvature prevents the per-sample free-prediction quadratic from becoming concave, but it cannot by itself guarantee neural-network optimization stability or cross-interval flow consistency.

E DNA PROMOTERS: ON-POLICY RL AND REWARD-GRADED DISTILLATION

The DNA experiments ask two questions. Does MFA work as on-policy RL for a generator whose state space is a manifold rather than a latent grid? And must the data targets come from the model's

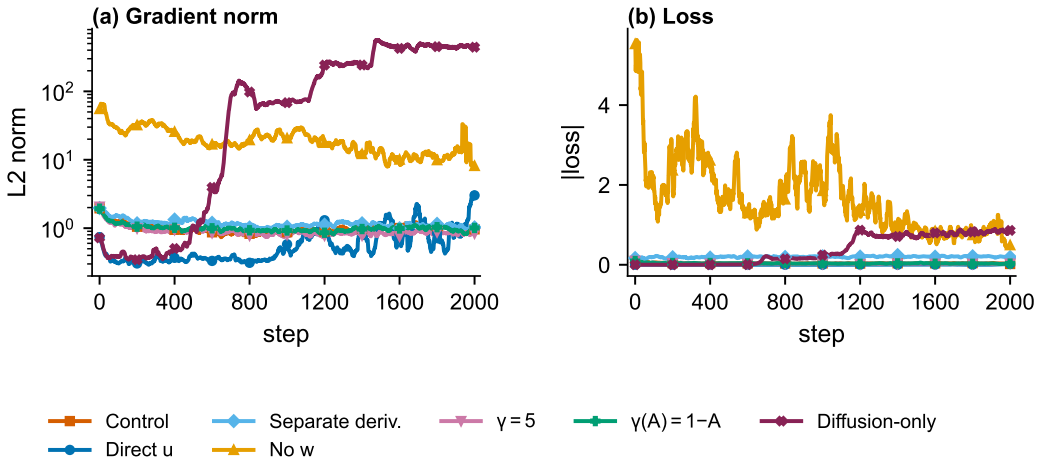


Figure S2: Optimization stability across MeanFlowAdvantage ablations, highlighting gradient norms, loss dynamics, and collapse behavior.

own rollouts at all? Nothing in Section 4 requires the latter: per input, the loss equation 10 needs only a target, a scalar weight, and an anchor.

Shared setup. We use a Riemannian MeanFlow (RMF) generator (Stark et al., 2024; Woo et al., 2026) pretrained locally on 1024-bp FANTOM5 promoters and conditioned on the regulatory signal; the reward is the negative Sei profile MSE (Chen et al., 2022). RMF places noise at time 0 and data at 1, so intervals run from t to $s \geq t$ and the average velocity lies in the tangent space of the sphere at x_t. We train on the training chromosomes, select checkpoints on 256 validation sequences (chr10) by Sei MSE, and report 512 test sequences (chr8–9), evaluated one-step with a fixed prior seed and hard one-hot decoding. The pretrained generator reaches 0.0714 Sei MSE and 0.928 6-mer correlation.

On-policy RL. MFA is used exactly as in Section 3, with no teacher. Per signal, the frozen rollout model draws $K = 8$ one-step sequences, Sei scores them, and equation 14 standardizes the rewards within the group, so a sequence below its group average gets $A < 0$. Each sequence is re-noised and the loss equation 10 is applied in the tangent space at x_t, with $\bar{u}$ and the shared correction from the rollout network as in RMF pretraining; intervals follow equation 15. MeanFlowNFT sees the same rollouts, advantages, intervals, and correction, differing only in the loss. Tables S2 and S3 give the configuration and per-seed numbers.

Two things stand out (Table S6, three seeds). First, teacher-free RL works on this manifold generator: MFA cuts the Sei MSE from 0.0714 to 0.0482 on average, and its best seed reaches 0.0437, below the distilled models below. Second, MFA and NFT are statistically indistinguishable on the training reward: all three paired intervals contain zero. This is what Proposition 2 predicts, since a rollout model that tracks the learner leaves only a gain difference between the two objectives. They differ in the statistic that is *not* optimized: MFA holds a 6-mer correlation of 0.953 ± 0.001 against 0.937 ± 0.009, in all three seeds. We report this as a consistent trend over three seeds, not a tested claim.

Reward-graded distillation. We now replace the on-policy target by a teacher, the setting of scientific design, where a strong but slow sampler exists and the goal is a one-step generator that keeps its gains. From the same noise, a frozen one-step rollout gives an endpoint b and a frozen ten-step Sei-guided teacher gives g. Their reward gap decides both whether a pair is used and how strongly:

$$\Delta r = r_{\mathrm{Sei}}(g) - r_{\mathrm{Sei}}(b), \qquad \text{keep the pair if } \Delta r \geq \tau, \qquad A = \mathrm{clip}(\eta\, \Delta r, 0, 1), \tag{37}$$

with $\tau = 0.002$ and $\eta = 10$. For each kept pair, the target is extrapolated past the teacher, and the rollout endpoint serves as both anchors:

$$y_\kappa = b + \kappa(g - b), \qquad \ell_{\mathrm{DNA}} = A\|y_\theta - y_\kappa\|_2^2 + (\gamma + \lambda)\|y_\theta - b\|_2^2, \tag{38}$$

with $\kappa = 1.5$ and $\gamma = \lambda = 0.01$. MeanFlowNFT receives the same pairs, advantages, extrapolation, learning rate, and validation protocol through its own loss, with $\beta = 0.8$ and reference weight $k = 0.01$.

Why graded advantages separate the two losses. As in equation 11, each pair has a closed-form optimum. For MFA and for NFT (via the expansion in Appendix A.4),

$$y^\star_{\mathrm{MFA}} - b = \frac{\kappa A}{A + \gamma + \lambda}\,(g - b), \qquad y^\star_{\mathrm{NFT}} - b = \frac{\kappa A}{\beta + k}\,(g - b). \tag{39}$$

MFA separates *where* a pair points from *how much* it counts: since $\gamma + \lambda = 0.02$, every accepted pair targets nearly the full extrapolated endpoint and A mainly sets the curvature $A + \gamma + \lambda$, that is, how hard the pair pulls. In NFT the advantage rescales the target itself, so pairs with a small but real improvement barely move while pairs with $A \approx 1$ overshoot to $1.85(g - b)$. With $A \equiv 1$ both reduce to a fixed extrapolation (Appendix E.1).

Table S6: One-step promoter generation on the test chromosomes ($n = 512$), from the same RMF initialization and the same protocol within each block. Sei MSE is the training reward; the 6-mer correlation is never optimized. On-policy rows are mean $\pm$ s.d. over three training seeds, with the best seed in brackets; distillation rows are the single reported run. Δ is the paired NFT$-$MFA Sei MSE with a 95% sequence bootstrap interval, so a positive Δ favors MFA.

Method	Sei MSE↓	6-mer corr.↑	Δ [95% CI]
Pretrained RMF	0.0714	0.928	—
On-policy RL, no teacher (3 seeds)			
MeanFlowNFT	**0.0471** $\pm$ 0.0048	0.937 $\pm$ 0.009	-0.0012 $[-0.0051, 0.0026]$
MeanFlowAdvantage	0.0482 $\pm$ 0.0051 [0.0437]	**0.953** $\pm$ 0.001	
Reward-graded distillation from a 10-step Sei-guided teacher			
MeanFlowNFT	0.0570	**0.955**	**0.0103** $[0.0064, 0.0143]$
MeanFlowAdvantage	**0.0467**	0.953	

Distillation results. Both methods select step 90 on validation. MFA reaches a Sei MSE of 0.0467, 18% lower than NFT on identical pairs and advantages, and here the gap is real: the paired difference is 0.0103 with a 95% interval of $[0.0064, 0.0143]$ (Table S6), and Figure S3 shows that it persists throughout training. Because that interval only resamples sequences, we repeated the whole comparison with four training seeds; Table S7 shows the gap survives training randomness as well. The contrast with the on-policy block is the point of this section. With graded teacher improvements the two objectives use the reward differently, as equation 39 makes explicit, and with $A \equiv 1$ they become indistinguishable again (Table S8). Distillation also loses the calibrated negative direction of on-policy RL: a teacher sample worse than the student does not say which direction is better. Unfiltered signed variants performed poorly (Appendix E.1), and we leave negative directions for distillation to future work.

E.1 Additional promoter results

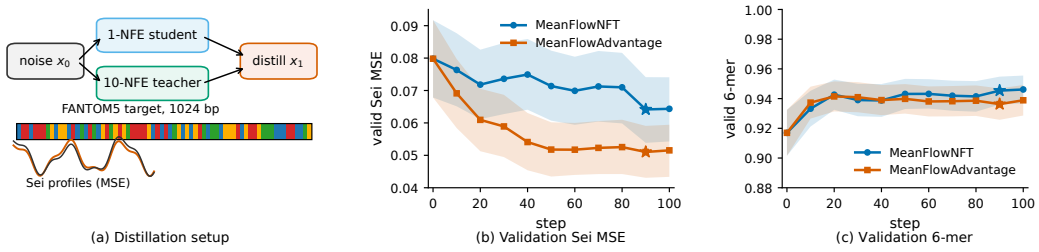


Figure S3: Reward-graded MeanFlow distillation. (a) A frozen rollout and a Sei-guided teacher define the reward gap Δr and the advantage A. (b)–(c) Validation curves ($n = 256$); bands are sequence-bootstrap 95% intervals, and stars mark the checkpoints selected by Sei MSE.

Training-seed repeats of the distillation gap. The interval quoted in Table S6 resamples test sequences from one training run, so it says nothing about how much the gap moves when training is repeated. We therefore reran the whole comparison, both losses and the full protocol, with four training seeds. Everything else is held fixed: the same pretrained generator, the same teacher, the pair filter $\tau = 0.002$, $\kappa = 1.5$, ten teacher steps, $\eta = 10$, and the same validation selection. Table S7 lists the result. MFA wins on every seed, and the spread across seeds is small next to the gap itself: the paired difference is 0.0115 with a seed-level standard deviation of 0.0011, and a bootstrap that resamples seeds and sequences together gives $[0.0091, 0.0138]$. The reduction quoted in the main text is the smallest of the four (18.1%); the four seeds average 19.7%. MFA is also the steadier of the two, varying by 0.0005 across seeds against 0.0010 for NFT. We did not sweep the remaining knobs of equation 37, so the filter threshold, the extrapolation κ, the teacher budget, and the reward scale η are fixed throughout at the values above; the $A \equiv 1$ rows of Table S8 vary κ and the learning rate only.

Table S7: Reward-graded distillation repeated over four training seeds. Checkpoints are selected on valid (n=256) and reported on test (n=512, eval seed 7). Δ is the paired NFT$-$MFA Sei MSE, with a 95% bootstrap over sequences; positive favors MFA. Seed 123 is the run reported in Table S6.

	MeanFlowNFT		MeanFlowAdvantage			
Seed	Sei MSE↓	6-mer↑	Sei MSE↓	6-mer↑	Δ [95% CI]	Red.
123	0.0570	0.955	**0.0467**	0.953	0.0103 [0.0064, 0.0143]	18.1%
42	0.0585	0.957	**0.0467**	0.953	0.0118 [0.0075, 0.0162]	20.2%
456	0.0586	0.958	**0.0477**	0.947	0.0110 [0.0068, 0.0152]	18.7%
789	0.0594	0.958	**0.0465**	0.949	0.0129 [0.0084, 0.0173]	21.7%
Mean	0.0584	0.957	**0.0469**	0.951	0.0115	19.7%
Std.	0.0010	0.001	**0.0005**	0.003	0.0011	1.6%

Resampling seeds and sequences jointly (4000 draws) gives a 95% interval of $[0.0091, 0.0138]$ for the mean paired difference, so the gap does not depend on the seed that was reported.

Constant-advantage special case. Setting $A \equiv 1$ discards the reward-gap magnitude after filtering. Because the rollout and reference anchors both equal b, the two DNA objectives then share an extrapolative regression form:

$$y_\theta - b = \frac{\kappa}{1+\gamma+\lambda}(g-b) \quad \text{(MFA)}, \qquad y_\theta - b = \frac{e}{\beta+k}(g-b) \quad \text{(MeanFlowNFT)}, \tag{40}$$

where e is the NFT target-extrapolation multiplier and k its reference weight. Table S8 shows comparable results when learning rate and nominal extrapolation are matched. The earlier individually tuned $A \equiv 1$ configurations are retained in Tables S9–S11 as ablations.

Table S8: $A \equiv 1$ special case under matched learning rate and nominal extrapolation. Checkpoints are selected on valid ($n = 256$) and evaluated on test ($n = 512$). Δ is paired NFT$-$MFA Sei MSE with a 95% bootstrap interval.

lr	κ	NFT MSE↓	MFA MSE↓	Δ [95% CI]
10^{-5}	1.25	0.0550	0.0545	$0.0005\ [-0.0007, 0.0018]$
10^{-5}	1.5	0.0490	0.0490	$0.0000\ [-0.0010, 0.0010]$
2.5×10^{-5}	1.25	0.0525	0.0518	$0.0007\ [-0.0013, 0.0028]$
2.5×10^{-5}	1.5	0.0474	0.0482	$-0.0009\ [-0.0025, 0.0008]$

Negative-direction ablation. In on-policy RL, negative advantages suppress sampled actions relative to the rollout policy. Here, a teacher-worse endpoint does not define a calibrated opposite target: unfiltered signed variants reach only 0.0744–0.0755 best valid Sei MSE, versus 0.0510 for filtered graded-advantage MFA. We therefore retain the filter and leave distillation-specific negative

directions for future work. Figure S4 collects the $A \equiv 1$ diagnostics: pretraining stability, paired valid differences, and evaluation-size sensitivity. Figure S5 shows training-pair teacher/student Sei MSE for the same special case. Table S11 varies the training seed for the earlier tuned $A \equiv 1$ configurations while fixing the RMF initialization and eval seed 7.

Table S9: Individually tuned $A \equiv 1$ special case on valid 1-NFE FANTOM5 ($n = 512$, chr10). Online selection used $n = 256$.

Method	step	Sei MSE↓	6-mer↑
Pretrained RMF	0	0.0754	0.924
MeanFlowNFT	70	0.0553	0.951
MeanFlowAdvantage	55	**0.0501**	0.950

Table S10: Paired valid Sei ΔMSE (NFT $-$ MFA) for the individually tuned $A \equiv 1$ special case, n=512, 2000 bootstrap draws. The interval is the 95% percentile CI.

step	ΔMSE	95% CI
0	0.000	[0.000, 0.000]
10	0.0103	[0.0065, 0.0141]
50	0.0065	[0.0031, 0.0101]
60	0.0058	[0.0025, 0.0092]
70	0.0052	[0.0022, 0.0083]
100	0.0081	[0.0044, 0.0116]

Table S11: FANTOM5 training-seed ablation for the individually tuned $A \equiv 1$ special case. All runs start from the same RMF `ckpts/0120` snapshot. Checkpoints are selected on valid (n=256, eval seed 7) and reported on test (n=512, eval seed 7). Only the training seed changes.

	MeanFlowNFT			MeanFlowAdvantage		
Seed	step	Sei MSE↓	6-mer↑	step	Sei MSE↓	6-mer↑
123[†]	70	0.0542	0.956	55	**0.0486**	0.958
42	60	0.0550	0.955	55	**0.0492**	0.955
456	100	0.0550	0.961	45	**0.0496**	0.951
789	90	0.0551	0.955	35	**0.0479**	0.953
Mean	—	0.0548	0.957	—	**0.0488**	0.954
Std.	—	0.0004	0.003	—	0.0007	0.003

[†]Original tuned $A \equiv 1$ run. Std. is the sample standard deviation over the four seeds.

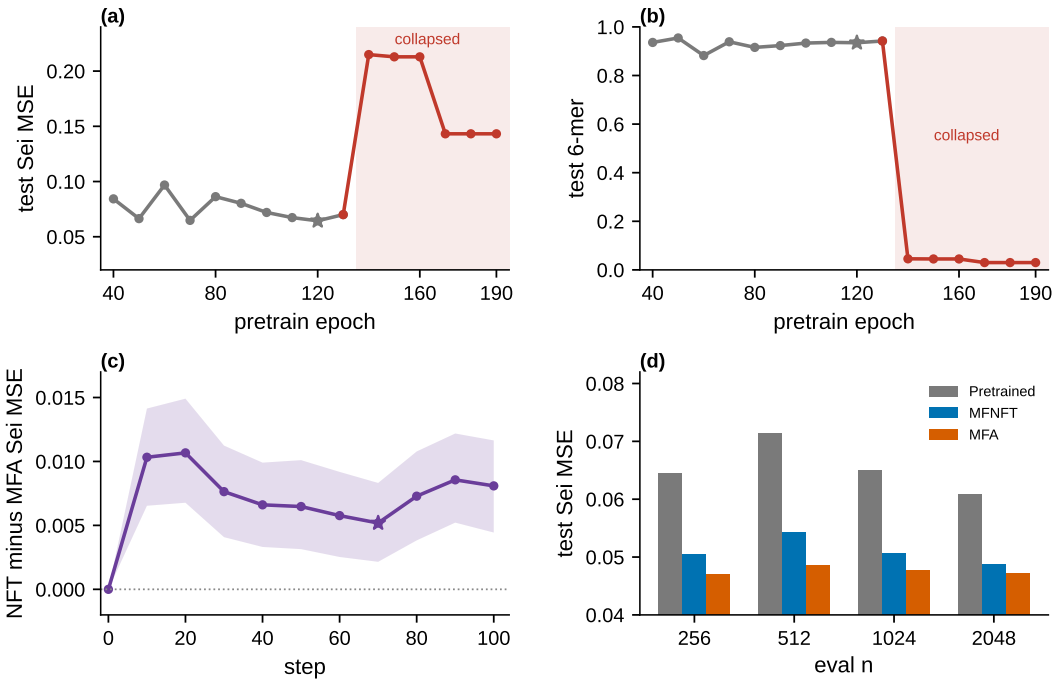


Figure S4: Promoter analyses that are not in Figure S3. (a)–(b) Test 1-NFE Sei MSE and 6-mer versus pretrain epoch (n=256). The star is the RL init (`ckpts/0120`); the shaded region is the collapsed tail. (c) Paired valid ΔMSE (NFT − MFA) with 95% sequence-level bootstrap CIs (n=512). The star is the on-grid NFT pick (step 70). (d) Test Sei MSE at the tuned $A \equiv 1$ checkpoints versus CAGE-sorted eval $n \in \{256, 512, 1024, 2048\}$.

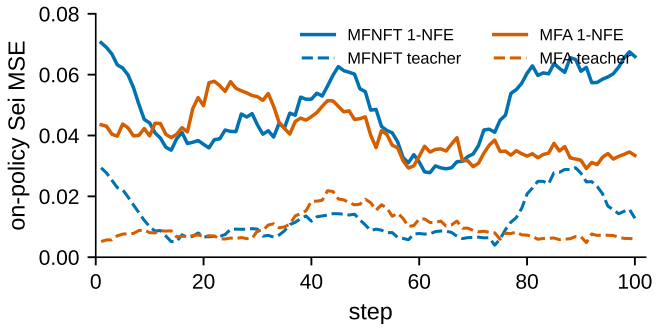


Figure S5: Training-pair Sei MSE of accepted distillation pairs (length-15 rolling mean). Solid: 1-NFE student. Dashed: 10-NFE teacher. These scores are computed for the accept filter ($\Delta r_{\text{Sei}} \geq 0.002$); Table S6 instead uses hard-decoded held-out sequences scored by Sei.