MEND: RL VIA PROXIMAL TARGET MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward-based post-training of flow models often relies on KL penalties, reference models, or thousands of updates. Direct reward backpropagation provides a direction but does not check whether each proposed change justifies its magnitude. We introduce MEND, a reinforcement learning method based on proximal target matching. A reward cap assigns unchanged targets to high-scoring samples. Below the cap, MEND proposes moves along the reward gradient and accepts a move only when its capped reward gain exceeds a quadratic displacement penalty. The model learns the accepted displacements through regression, without a KL term, frozen reference model, or advantage weights. MEND scores higher than the Flow-GRPO adapter on five of six evaluators at the same reported distance to base-model images. Under an equal-budget protocol, it exceeds the reported ReFL and DiffusionNFT curves at every reported update across four training rewards, reaching PickScore 24.03 versus 23.92 and 23.43, respectively. A three-reward run also exceeds the released DiffusionNFT checkpoint on all three training rewards. The method transfers to another flow backbone and a distilled model. Code and checkpoints will be released.

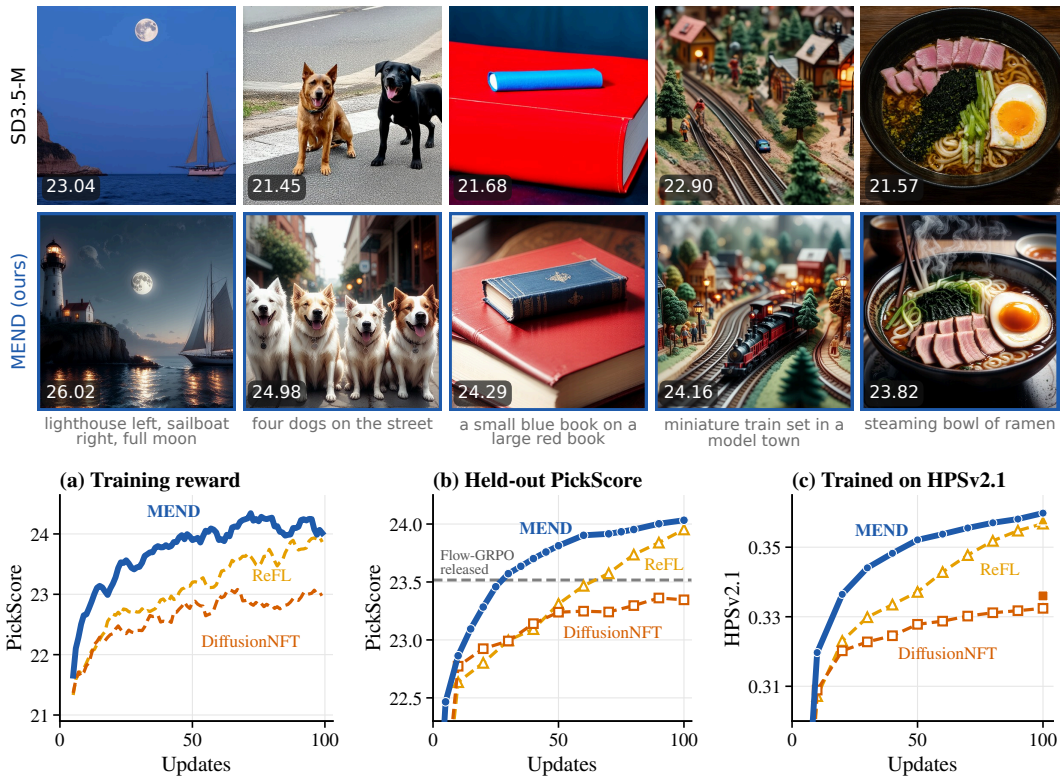

Figure 1: **MEND stays ahead of the reported baselines at every update.** Top: SD3.5-M above, MEND below, same prompt and initial noise. Bottom, equal budget: (a) training reward and (b) held-out PickScore against the reported ReFL and DiffusionNFT curves, with the released adapters at their own settings as levels; (c) the HPSv2.1 run's trained reward.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

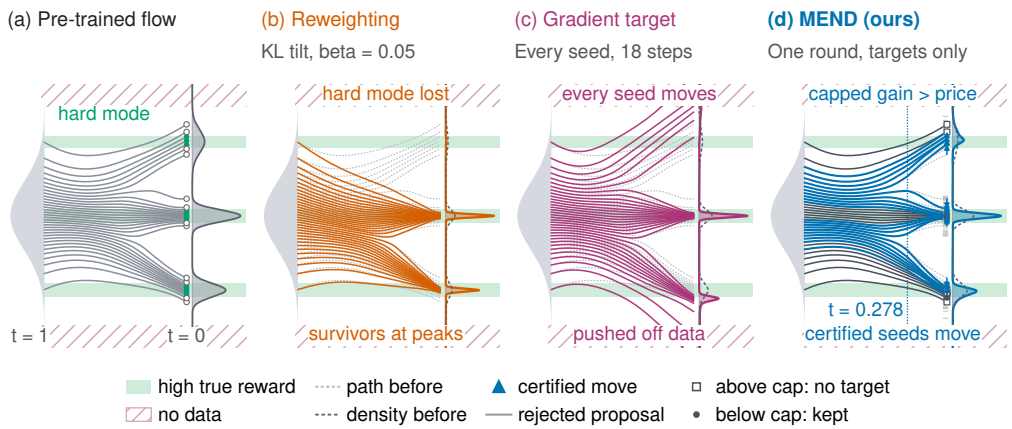

Figure 2: (a) The base flow. (b) Reward reweighting concentrates on the peaks and loses the hard mode. (c) Gradient ascent moves every sample and pushes some of the mass off the data. (d) MEND: capped samples stay, certified moves are short, and the hard mode keeps its mass (Appendix F.3).

1 Introduction

Learned rewards enable text-to-image flow models to be post-trained for human preferences (Kirstain et al., 2023; Wu et al., 2023; Xu et al., 2023). A reward score, however, does not specify which samples should change or where they should move. Flow-GRPO favors high-scoring samples through group-relative policy gradients, while DiffusionNFT uses reward-weighted regression (Liu et al., 2025; Zheng et al., 2026). The released checkpoints we compare were trained for thousands of updates.

Reward backpropagation provides a direction: ReFL and DRaFT differentiate rewards through sampling (Xu et al., 2023; Clark et al., 2024). But a gradient does not check whether a proposed change earns its displacement cost, and high-scoring samples can still receive updates. A KL penalty or reference term limits distributional drift without making this per-sample decision. In the three-mode example of Figure 2, reweighting suppresses a hard mode, while unchecked ascent pushes samples into low-density regions.

Table 1: **Objective components.** Blue matches MEND; red differs. The check compares reward gain with displacement cost. MEND uses a lagged behavior adapter, not a frozen reference. Updates refer to the compared runs.

Method	KL / frozen reference	Reward weights	∇R	Target check	Updates
Flow-GRPO	yes	yes	no	no	∼4k
ReFL	no	no	yes	no	100
DiffusionNFT	yes	yes	no	no	1.7k
MEND	no	no	yes	yes	100

We introduce MEND, an RL method based on *proximal target matching* (Figure 4). Within each prompt group, it caps rewards and proposes local moves along the normalized reward gradient for samples below the cap. It selects the candidate with the highest capped reward minus a quadratic displacement penalty, with the unchanged sample always included. Regression then matches the behavior prediction plus the selected displacement at a stored rollout state, without a KL term, frozen reference model, or advantage weights (Table 1). Every accepted target improves capped reward by more than its quadratic penalty, and its displacement bound shrinks to zero as the sample approaches the cap. At the behavior parameters, the regression descent direction backpropagates normalized endpoint reward gradients through accepted samples' clean predictions, scaled by their selected steps. These statements concern a frozen round of targets, not the optimizer step or a training run; model fidelity is measured separately (Section 4.3).

After 100 PickScore updates, MEND scores higher than the released Flow-GRPO adapter on every evaluator except aesthetic score, at the same reported distance to base-model images (Table 2). Under the equal-budget protocol of Zhou et al. (2026), it exceeds the reported ReFL and DiffusionNFT curves at every reported update across four training rewards (Figures 1 and 8). Figures 1 and 3 show selected comparisons. Contributions:

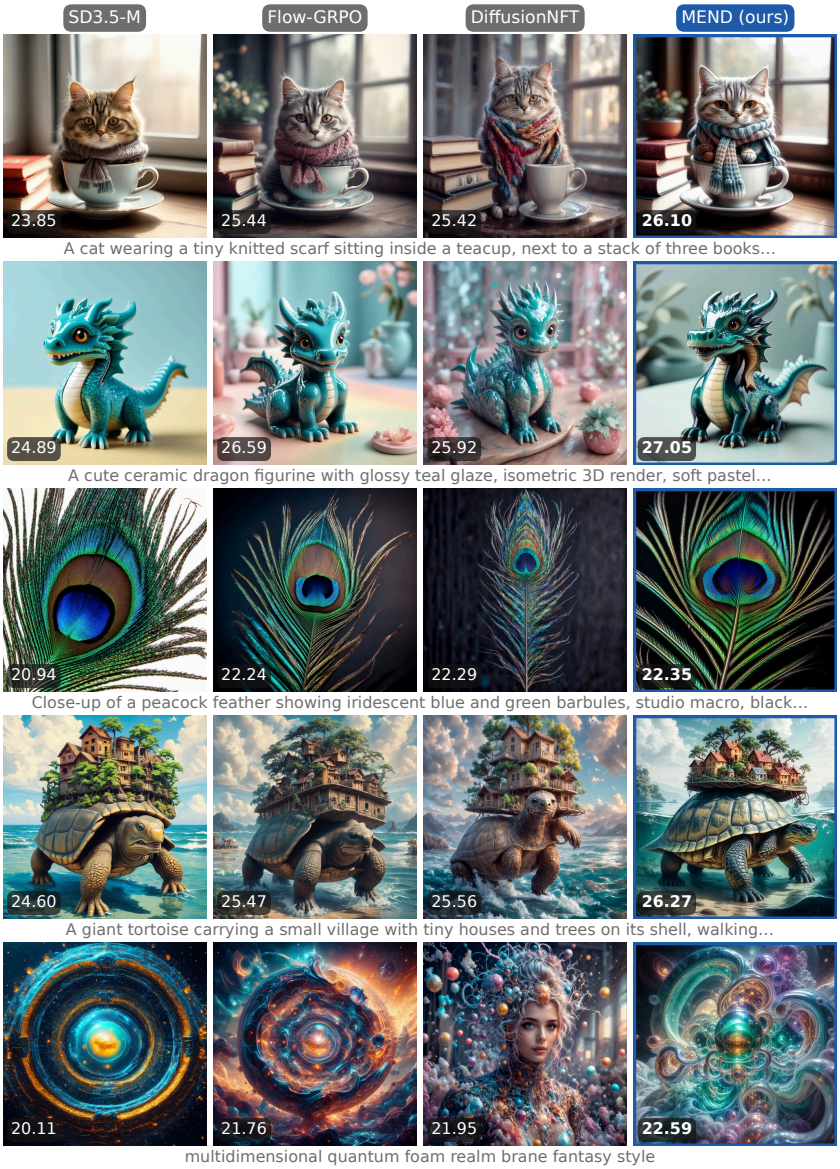

Figure 3: Rows share a prompt and seed across SD3.5-M, Flow-GRPO, DiffusionNFT, and MEND. PickScore on each tile.

- **Proximal target matching.** MEND selects local targets by balancing capped reward gain against a quadratic displacement penalty, then learns the selected changes through regression.
- **Target-level guarantees.** We bound accepted displacements by capped reward gains and characterize the regression gradient at the behavior parameters.
- **Update efficiency and transfer.** In 100 updates, MEND scores higher than released Flow-GRPO (about 4k updates) on five of six evaluators and reported ReFL and DiffusionNFT on four rewards at equal budget. The method transfers to two other backbones.

2 METHOD

Setting and notation. A rectified flow uses the interpolation $z_t = (1-t)x_0 + t\epsilon$ between a clean latent x_0 and Gaussian noise ϵ. For prompt c, its velocity $v_\theta(z_t, t, c)$ gives the clean prediction $\hat{x}_\theta(z_t) = z_t - t\, v_\theta(z_t, t, c)$ (Liu et al., 2022; Lipman et al., 2022; Esser et al., 2024). Two adapters share the base model: the trained adapter θ and the behavior adapter θ_{old}. For each prompt, θ_{old} generates G deterministic 10-step trajectories (Lu et al., 2022), with endpoints $x^{(1)}, \ldots, x^{(G)}$ and

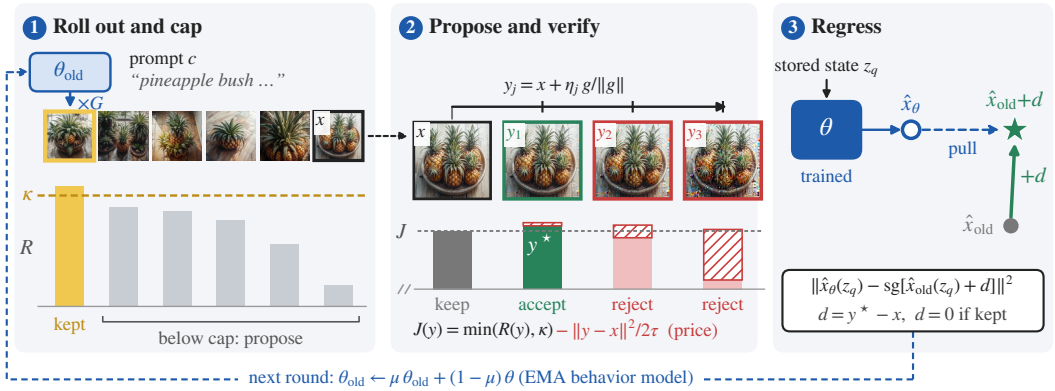

Figure 4: MEND overview. (1) A group for prompt c; samples at or above the cap κ are kept. (2) Each sample below the cap gets 3 proposals along its reward gradient; the verdict subtracts a distance price from the capped reward and keeps the best candidate, here the shortest move y_1. (3) The trained adapter regresses at the stored state onto the behavior prediction plus the accepted displacement d; the behavior model then follows it by EMA.

stored states z_q nearest $t_q = 0.278$. We use the root-mean-square norm $\|\cdot\|$ and its mean inner product $\langle\cdot,\cdot\rangle$. $R(x, c)$ scores the decoded latent, and $g = \nabla_x R(x, c)$ is its gradient under this inner product. Each round consists of a rollout, target selection (cap, propose, verify), and one regression update (Algorithm 1).

Cap. Within each prompt group, we cap rewards at the quantile $q = 0.75$, subject to a global floor κ_{glob} that rises slowly across rounds (Appendix B):

$$\kappa(c) = \max\Big\{Q_q\big(\{R(x^{(k)}, c)\}_{k=1}^{G}\big),\ \kappa_{\text{glob}}\Big\}, \qquad R_\kappa(x, c) = \min\{R(x, c), \kappa(c)\}. \tag{1}$$

Samples at or above the cap receive zero displacement.

Propose. Each sample below the cap with $g \neq 0$ receives $K = 3$ proposals along its normalized reward gradient, each decoded and scored once:

$$y_j = x + \eta_j \frac{g}{\|g\|}, \qquad \eta_j \in \{0.1, 0.2, 0.4\}. \tag{2}$$

Samples with $g = 0$ are kept.

Verify. We select the candidate with the highest capped reward minus a quadratic displacement penalty, always including the unchanged sample:

$$y^\star = \arg\max_{y \in \{x, y_1, \ldots, y_K\}} J(y), \qquad J(y) = R_\kappa(y, c) - \frac{\|y - x\|^2}{2\tau}. \tag{3}$$

Ties favor x, then the shortest proposal. The parameter $\tau > 0$ is fixed within a round and adjusted between rounds to target an acceptance rate of 30% to 60% among proposed samples (Appendix B).

Regress. Let $d = y^\star - x$ for accepted samples and $d = 0$ otherwise. At the stored state z_q, we match the behavior prediction plus this displacement, rather than the endpoint $y^\star$ itself:

$$\mathcal{L}(\theta) = \mathbb{E}_{\text{moved}} \|\hat{x}_\theta(z_q) - \text{sg}[\hat{x}_{\text{old}}(z_q) + d]\|^2 + \lambda_{\text{keep}}\, \mathbb{E}_{\text{kept}} \|\hat{x}_\theta(z_q) - \text{sg}[\hat{x}_{\text{old}}(z_q)]\|^2. \tag{4}$$

Here $\hat{x}_{\text{old}} = \hat{x}_{\theta_{\text{old}}}$, sg stops gradients, and $\lambda_{\text{keep}} = 10$. Each expectation averages its subset of the round; an empty subset contributes zero.

Behavior model and optimizer. The behavior adapter follows θ with EMA coefficient $\mu_u = \min(0.001\, u, 0.5)$ after update u (Algorithm 1, line 7). The keep term therefore anchors predictions to a lagged adapter, not a frozen reference. Each round takes one AdamW step with constant learning rate 3×10^{-4} and $\epsilon = 10^{-12}$; evaluation uses a separate EMA of θ (Algorithm 2). Reward computation requires one backward pass per sample, plus 3 additional decodes and reward evaluations per proposed sample.

Algorithm 1 MEND: one training round.

Require: adapters θ, θ_{old}; penalty scale τ; cap quantile q; steps $\eta_1, \ldots, \eta_K$

1: Roll out θ_{old} for every prompt and seed; store endpoint x and state z_q ▷ rollout
2: Compute $R(x, c)$ and g for all samples; set $\kappa(c)$ by equation 1 ▷ cap
3: **for** each sample with $R(x, c) < \kappa(c)$ and $g \neq 0$ **do**
4: Form proposals by equation 2; select $y^\star$ by equation 3; $d \leftarrow y^\star - x$ ▷ propose, verify
5: **end for**
6: Set $d \leftarrow 0$ for all other samples; take one AdamW step on equation 4 ▷ regress
7: Update τ from the acceptance rate; $\theta_{\text{old}} \leftarrow \mu_u \theta_{\text{old}} + (1 - \mu_u)\theta$ ▷ penalty, behavior

3 Theoretical analysis

Target selection evaluates a proximal objective (Rockafellar, 1976; Jordan et al., 1998) over a finite candidate set containing the unchanged sample. Proofs are in Appendix C.

Proposition 1 (Verified target selection)**.** Fix a round's samples x, caps κ, price $\tau > 0$, and proposal steps $0 < \eta_1 \leq \cdots \leq \eta_K$. Include x among the candidates and accept a move only if its objective strictly exceeds that of x. Every accepted target satisfies

$$\frac{\|y^\star - x\|^2}{2\tau} \leq R_\kappa(y^\star) - R_\kappa(x) \leq \kappa - R_\kappa(x). \tag{5}$$

Hence $\|y^\star - x\| \leq \min\{\eta_K, \sqrt{2\tau(\kappa - R_\kappa(x))}\}$ and $R(y^\star) - R(x) \geq \|y^\star - x\|^2/(2\tau)$.

Proposition 2 (First-order realization)**.** Fix a round's targets and rollout states, and let $\hat{x}_\theta$ be differentiable in θ. Let $\mathcal{A} \neq \varnothing$ be the accepted seeds, with steps η_i, normalized endpoint reward gradients $\hat{h}_i = g_i/\|g_i\|$, and stored states $z_q^{(i)}$. Then

$$-\nabla\mathcal{L}(\theta_{\text{old}}) = \frac{2}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \eta_i \; \nabla_\theta \big\langle \hat{h}_i, \, \hat{x}_\theta(z_q^{(i)}) \big\rangle \Big|_{\theta = \theta_{\text{old}}}, \tag{6}$$

so kept seeds contribute zero gradient at θ_{old}.

Equation equation 6 backpropagates normalized endpoint reward gradients through the clean predictions, restricted to accepted seeds and scaled by their selected steps. This identity holds at the behavior parameters; it does not describe an AdamW step or a full training run.

4 Experiments

4.1 Setup

We train SD3.5-M (Esser et al., 2024) on PickScore under the equal-budget protocol of Zhou et al. (2026): 48 Pick-a-Pic prompts (Kirstain et al., 2023) $\times$ 24 images per update, rank-32 LoRA, 10-step rollouts at 512 pixels, and 100 updates. ReFL and DiffusionNFT results under this protocol are quoted from that work. Evaluation uses DrawBench (Saharia et al., 2022), 200 prompts $\times$ 5 seeds and 40 Euler steps, scored with PickScore, HPSv2.1 (Wu et al., 2023), HPSv3 (Ma et al., 2025), ImageReward (Xu et al., 2023), CLIPScore (Hessel et al., 2021) and aesthetic quality (Schuhmann et al., 2022). Fidelity is the DreamSim distance (Fu et al., 2023) to the base image generated with the same prompt, noise and sampling settings. We re-evaluate the released Flow-GRPO PickScore adapter and DiffusionNFT multi-reward checkpoint in this pipeline. Design decisions used the first 64 DrawBench prompts; Appendix D.1 reports results on the disjoint remainder. Main comparisons use one training seed per configuration.

4.2 Main comparison

Table 2 compares released checkpoints with two MEND runs at the released adapters' sampling settings, as do Figures 5 to 7 except Figure 6(a). Table 3 and Figures 1 and 8 use the equal-budget protocol's setting (Appendix B). The PickScore run matches Flow-GRPO's base distance at displayed precision, with a roughly 40-fold reduction in updates. Aesthetic score is the only evaluator on which it scores lower. The run trained on PickScore, HPSv2.1 and CLIPScore exceeds the released

Table 2: MEND exceeds Flow-GRPO on five of six evaluators at the same displayed base distance. Its three-reward run exceeds DiffusionNFT on those three rewards. DrawBench 200 × 5.

Method	Rewards	Updates	PickScore	HPSv2.1	HPSv3	Image Reward	CLIPScore	Aesthetic	Dist. to base ↓
SD3.5-M	none	0	22.35	0.280	2.72	0.83	0.283	5.39	0
Flow-GRPO	PickScore	∼4k	23.52	0.316	7.04	1.27	0.280	5.90	**0.313**
DiffusionNFT	five	1.7k	<u>23.82</u>	<u>0.331</u>	**7.50**	**1.49**	<u>0.292</u>	**6.02**	0.538
MEND (ours)	PickScore	100	23.70	0.319	<u>7.15</u>	<u>1.32</u>	0.291	5.88	**0.313**
MEND (ours)	three	300	**23.89**	**0.344**	5.87	1.28	**0.300**	<u>5.92</u>	<u>0.488</u>

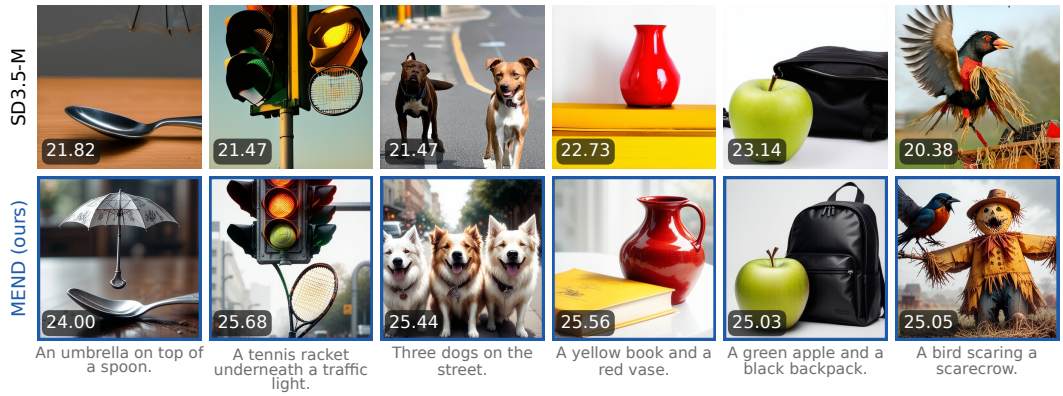

Figure 5: DrawBench prompts (text under each column), SD3.5-M above and MEND below from the same initial noise, PickScore on each tile.

DiffusionNFT checkpoint (1.7k updates on five rewards) on those three evaluators at a smaller base distance, but scores lower on rewards trained only by DiffusionNFT. Section 4.3 compares equal budgets. Figure 5 shows selected DrawBench examples where MEND renders counts, attributes or spatial relations missed by SD3.5-M from the same noise.

4.3 Update efficiency and fidelity

Under the equal-budget protocol, Figure 1(b) plots evaluation PickScore against updates. MEND reaches 24.03 at update 100, compared with the reported 23.92 for ReFL and 23.43 for DiffusionNFT, and remains above both curves at every reported update. On 64 unseen Pick-a-Pic prompts (Figure 6(a)), it exceeds our DiffusionNFT run under the same protocol at every evaluated update. It also exceeds the released Flow-GRPO and DiffusionNFT scores by update 25; those checkpoints used about 4k and 1.7k updates, respectively. Training costs 10.0 GPU-hours on 3 GB200 GPUs, excluding evaluation. Reward continues to improve while base distance changes little late in training: between updates 25 and 100, DrawBench PickScore rises from 23.36 to 23.70, while base distance changes from 0.294 to 0.313 (Table 2, Figure 6(c)). Figure 7 follows two seeds through this run. This fidelity is measured, not guaranteed by the target check.

4.4 Other rewards and backbones

Table 3, Figure 8 and Figure 1(c) compare separate runs trained on PickScore, ImageReward, HPSv2.1 or CLIPScore under the equal-budget protocol. Each run exceeds the reported ReFL and DiffusionNFT scores on its training reward at update 100 and at every reported update. All four runs finish above the base on every held-out evaluator, although three show held-out declines before update 100 (Appendix E.1). CLIPScore and ImageReward training yield the lowest held-out PickScore (22.27 and 22.33); Appendix E.3 examines the declines under longer training.

On SD3-M, PickScore rises from 20.36 to 23.70 (Table 8), exceeding the released Linear-DPO checkpoint at 20.96, while held-out HPSv2.1 rises from 0.215 to 0.298. On Z-Image-Turbo (Z-Image Team et al., 2025), a distilled model sampled in nine steps at 1024 pixels, PickScore training raises that score from 22.86 to 24.10 and held-out HPSv2.1 from 0.295 to 0.315. HPSv2.1 training

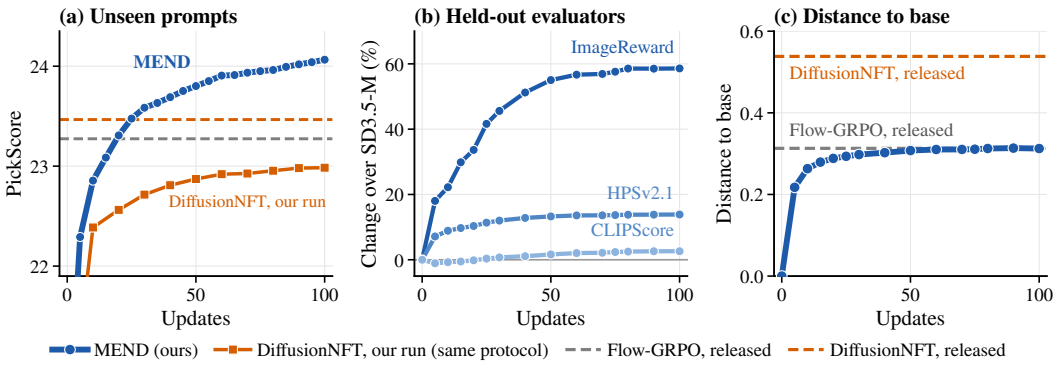

Figure 6: (a) PickScore on 64 held-out Pick-a-Pic prompts at the protocol's setting, against our DiffusionNFT run under the same protocol, with the released checkpoints as levels; (b) held-out evaluators and (c) distance to the base on DrawBench at the released adapters' setting.

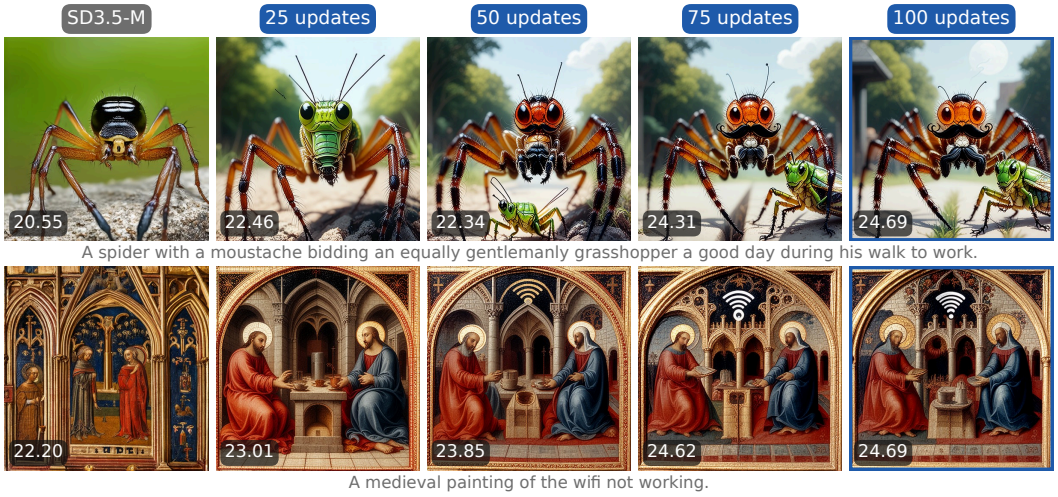

Figure 7: SD3.5-M and MEND after 25, 50, 75 and 100 updates, with the same initial noise along each row and PickScore on each tile. Top: the grasshopper appears by update 50 and the moustache by update 75. Bottom: the wifi symbol appears at update 50 and is sharp by update 75.

reaches 0.357 with held-out PickScore 23.27. All runs use the same update budget; Appendix E.1 gives full results (Tables 8 and 11) and training trajectories.

4.5 Ablations

We vary one component at the development budget (16 prompts $\times$ 8 images per update, 50 updates, the earlier optimizer setting). Table 12 in Appendix F evaluates the checkpoints on full DrawBench; Figure 9 uses the development split, where the fixed-step control is available. MEND accepts target moves for 34% of seeds per round, reaching PickScore 23.10 at diversity 0.297. Removing the cap or price increases this share to 45% and 78%, respectively, and lowers diversity to 0.287 and 0.291. Assigning a move to every seed below the cap (81%) yields the highest full-set PickScore, 23.45, and lowest diversity, 0.260. At the full per-update budget, varying the initial price, using one candidate or removing the behavior EMA changes update-50 PickScore by at most 0.06; five candidates lower it by 0.25 (Appendix F.2).

4.6 Qualitative results

Figures 1, 3, 5, 7, 10 and 11 use matched prompts and initial noise, with PickScore on each tile. Figures 3 and 11 compare MEND with the released models; Figure 10 shows base-to-MEND pairs. Examples are selected by eye, whereas tables use the complete evaluation sets. Many MEND exam-

Table 3: Each MEND run uses SD3.5-M, 100 updates and the protocol of Zhou et al. (2026); the base uses the same evaluation setting. The last two columns give baseline scores on the row's training reward. Bold: MEND's trained metric; underline: second best among the three methods on that metric.

Training reward	PickScore	HPSv2.1	ImageReward	CLIPScore	ReFL (reported)	DiffusionNFT (reported)
none (SD3.5-M)	20.58	0.207	−0.52	0.239		
PickScore	**24.03**	0.301	1.13	0.273	<u>23.92</u>	23.43
ImageReward	22.33	0.303	**1.51**	0.266	1.28	<u>1.46</u>
HPSv2.1	23.06	**0.360**	1.21	0.268	<u>0.358</u>	0.336
CLIPScore	22.27	0.268	0.98	**0.314**	<u>0.308</u>	0.298

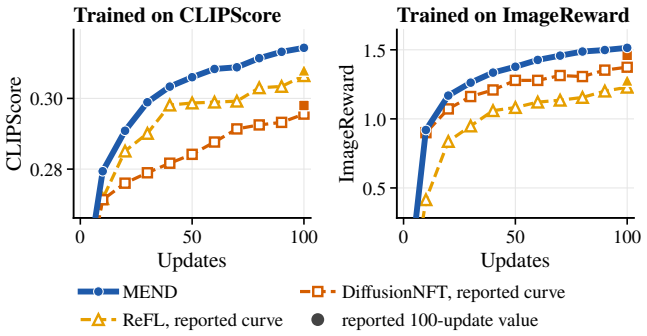

Figure 8: Trained reward against updates under the equal-budget protocol of Zhou et al. (2026), one MEND run per reward, against the reported ReFL and DiffusionNFT curves and 100-update values.

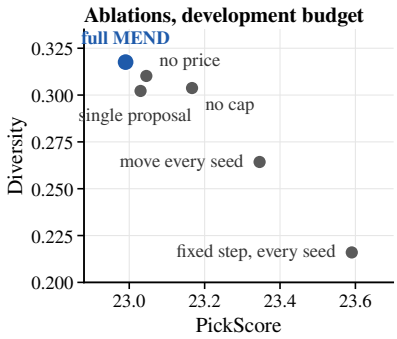

Figure 9: **Restricting moves trades reward for diversity.** One-factor arms at the development budget on the development split, with the fixed-step control; single training seed.

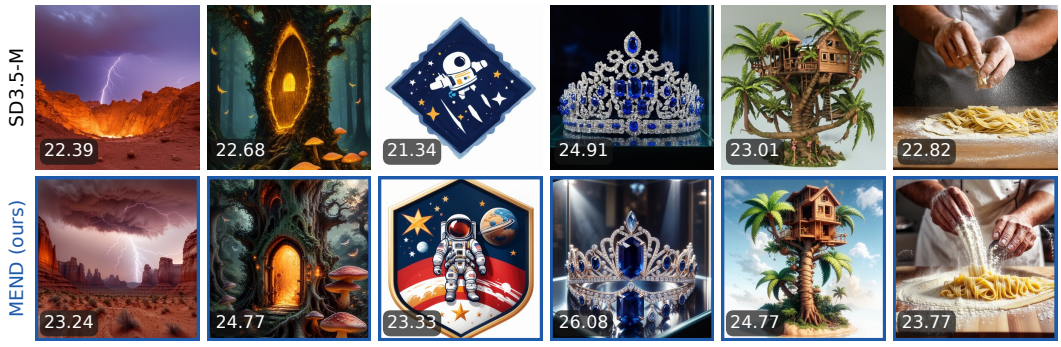

Figure 10: More SD3.5-M and MEND results.

ples have higher saturation, reflecting a reward preference examined in Appendix E.3. Appendix G gives full prompts, additional seeds and training progressions.

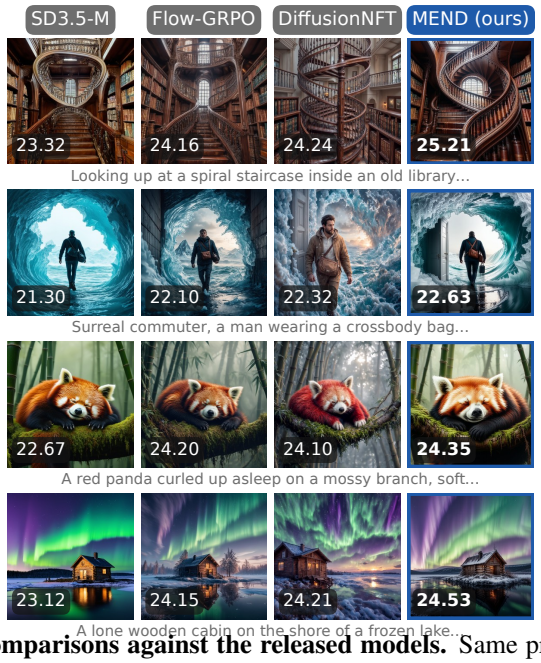

Figure 11: **Matched comparisons against the released models.** Same prompt and seed per row; PickScore on each tile, best in bold.

5 Related Work

DDPO and DPOK optimize rewards with policy gradients, while Diffusion-DPO fits offline preferences (Black et al., 2024; Fan et al., 2023; Wallace et al., 2024). Flow-GRPO and DanceGRPO extend group-relative policy gradients to flow models with stochastic sampling and a KL penalty (Liu et al., 2025; Xue et al., 2025). DiffusionNFT, AWM and FlowAWR use reward- or advantage-weighted regression on model-generated samples (Zheng et al., 2026; Xue et al., 2026; Fu et al., 2026). These methods adjust the contribution of existing samples; MEND instead constructs and checks local targets (Appendix H).

ReFL, DRaFT and AlignProp backpropagate differentiable rewards through sampling (Xu et al., 2023; Clark et al., 2024; Prabhudesai et al., 2023); RAFT and related methods select among existing samples (Dong et al., 2023; Lee et al., 2026). Concurrently, Zhou et al. (2026) regress clean predictions toward bounded reward-gradient targets using group weights, while Self-OPD distills the model's own stochastic branches (Zhang et al., 2026). MEND accepts a proposal only when its capped reward gain over the unchanged sample exceeds a quadratic displacement penalty. This per-target check differs from distribution-level Wasserstein regularization (Fan et al., 2025; Hwang et al., 2026).

6 Conclusion

MEND post-trains flow models through proximal target matching: it accepts local changes only when their capped reward gain exceeds a quadratic displacement penalty, then learns the accepted displacements through regression. After 100 updates, it scores higher than the released Flow-GRPO adapter on every evaluator except aesthetic score, at the same reported distance to base-model images. The method transfers across rewards and backbones.

Limitations

MEND requires a differentiable reward and inherits its blind spots. The target-level guarantee applies to a frozen round, not to the optimizer step or a full training run. Each configuration uses one training seed, and development prompts overlap the evaluation set (Appendix D.1). The cap can delay reward hacking but does not prevent it: longer training exploits the reward beyond the 100-update budget (Appendices E.3 and F.4), while some reward-specific runs show declining scores on held-out evaluators even within this budget.

Ethics statement

Learned rewards reflect their annotators' preferences and can reward artifacts, so we report held-out evaluators and distance to the base alongside the training reward. Released weights will retain the base models' terms of use and safety filters.

Reproducibility statement

Algorithm 1 specifies the training round. Appendix B gives the training and evaluation settings and the baseline sources, and Appendix C gives the proofs. We will release code, checkpoints, evaluation images, per-prompt scores and the disjoint evaluation prompts.

Statement on the use of large language models

We used ChatGPT for English editing and text formatting, for editing sentences and paragraphs, for light restructuring of a few sections, and for a small amount of code that renders the result figures. It was not used to generate the research idea. We reviewed and verified every passage and every line of code it touched, and we take full responsibility for the final content of this work.

References

Gautham Govind Anil, Shaan Ul Haque, Nithish Kannen, Dheeraj Nagaraj, Sanjay Shakkottai, and Karthikeyan Shanmugam. Fine-tuning diffusion models via intermediate distribution shaping. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.02692.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, 2024. arXiv:2302.07121.

Yuchen Bao, Chao Wen, Haowei Wang, et al. ReNFT: Repairing mode collapse in reward post-training via internal probability-mass recalibration. *arXiv:2609.00061*, 2026.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.13301.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. arXiv:2209.14687.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.17400.

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 *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2409.08861.

Hanze Dong, Wei Xiong, Deepanshu Goyal, et al. RAFT: Reward ranked FineTuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. arXiv:2304.06767.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2403.03206.

Jiajun Fan, Shuaike Shen, Chaoran Cheng, et al. Online reward-weighted fine-tuning of flow matching with Wasserstein regularization. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2502.06061.

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 (NeurIPS)*, 2023. arXiv:2305.16381.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.09344.

Zheming Fu, Ruizhe He, Wei Shang, et al. FlowAWR: Online adaptive flow reinforcement via advantage-weighted rectification. *arXiv:2606.30376*, 2026.

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 (NeurIPS), Datasets and Benchmarks Track*, 2023. arXiv:2310.11513.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J. Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *International Conference on Learning Representations*, 2024. arXiv:2311.16424.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Hoseong Hwang, Woorim Han, Joungin Chun, et al. Reward-guided fine-tuning of one-step generative models via Wasserstein gradient flow. *arXiv:2608.29647*, 2026.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, et al. Distribution matching distillation meets reinforcement learning. In *European Conference on Computer Vision (ECCV)*, 2026a. arXiv:2511.13649.

Zhou Jiang, Yandong Wen, and Zhen Liu. Drifting preference optimization for one-step generative models. *arXiv:2606.02521*, 2026b.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker–Planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998.

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 (NeurIPS)*, 2023. arXiv:2305.01569.

Jaewoo Lee, Minsu Kim, Sanghyeok Choi, Inhyuck Song, Sujin Yun, Hyeongyu Kang, Woocheol Shin, Taeyoung Yun, Kiyoung Om, and Jinkyoo Park. Diffusion alignment as variational expectation-maximization. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.00502.

Kesong Li, Yixuan Xu, Kuo-kun Tseng, Weiyi Lu, Kan Liu, and Tao Lan. Linear-DPO: Linear direct preference optimization for diffusion and flow-matching generative models. *arXiv:2605.21123*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv:2210.02747*, 2022.

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:2505.05470*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv:2209.03003*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv:2211.01095*, 2022.

Yuhang Ma, Yunhao Shui, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2508.03789.

David McAllister, Miika Aittala, Tero Karras, et al. Finite difference flow optimization for RL post-training of text-to-image models. *arXiv:2603.12893*, 2026.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2108.01073.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv:2310.03739*, 2023.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

Chitwan Saharia, William Chan, Saurabh Saxena, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2205.11487.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks*, 2022.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. arXiv:2311.12908.

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: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 (NeurIPS)*, 2023. arXiv:2304.05977.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2509.25050.

Zeyue Xue, Jie Wu, Yu Gao, et al. DanceGRPO: Unleashing GRPO on visual generation. *arXiv:2505.07818*, 2025.

Z-Image Team, Huanqia Cai, Sihan Cao, et al. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv:2511.22699*, 2025.

Shiyi Zhang, Mushui Liu, Yunze Tong, Wanggui He, Siyu Zou, Jinlong Liu, Yunlong Yu, Jian Song, Hao Jiang, Pipei Huang, and Bo Zheng. Self-OPD: On-policy distillation for flow matching models without teacher. *arXiv:2608.26872*, 2026.

Kaiwen Zheng, Huayu Chen, Haotian Ye, et al. DiffusionNFT: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.16117.

Wei Zhou, Xiongwei Zhu, Lingdong Kong, et al. On-policy self-distillation in diffusion models. *arXiv:2608.24646*, 2026.

A ALGORITHM AS IMPLEMENTED

Algorithm 2 is Algorithm 1 with the bookkeeping of the implementation: the global cap floor, the tie rules, the price controller and the two EMAs. Notation follows Section 2; Q_q is the q-quantile of a set, u the update index and a the acceptance share. Rewards, cap statistics and loss normalizers are pooled over devices.

Algorithm 2 MEND: one training round, as implemented.

Require: adapters θ, θ_{old}, θ_{ema}; price τ; floor κ_{glob}; update index u

Roll out θ_{old} for every prompt c and seed; store the endpoint x and the state z_q ▷ rollout
$R(x, c)$ and $g = \nabla_x R(x, c)$ for every sample, in one backward pass ▷ score
Raise κ_{glob} toward the round's median reward; never lower it ▷ floor
$\kappa(c) \leftarrow \max\{Q_q(\{R(x^{(k)}, c)\}_k),\ \kappa_{\text{glob}}\}$ for every prompt ▷ cap
for each sample with $R(x, c) < \kappa(c)$ **do**
    $y_j \leftarrow x + \eta_j\, g/\|g\|$ for $j = 1, \ldots, K$ ▷ propose
    $y^\star \leftarrow \arg\max_{y \in \{x, y_1, \ldots, y_K\}} R_\kappa(y, c) - \|y - x\|^2/(2\tau)$; ties to x, then to the shortest ▷ verify
end for
$d \leftarrow y^\star - x$ for moved samples; $d \leftarrow 0$ for every other sample
One AdamW step on $\mathcal{L}(\theta)$ of equation 4; $u \leftarrow u + 1$ ▷ regress
$\theta_{\text{ema}} \leftarrow \nu_u \theta_{\text{ema}} + (1 - \nu_u)\theta$ ▷ evaluated weights
$a \leftarrow$ share of proposed samples that moved; rescale τ if a leaves its band; clamp ▷ price
$\theta_{\text{old}} \leftarrow \mu_u \theta_{\text{old}} + (1 - \mu_u)\theta$ ▷ behavior

B EVALUATION SETTINGS

Protocols. Protocol O follows Zhou et al. (2026, Appendix C.1): SD3.5-M with a LoRA adapter, 48×24 Pick-a-Pic images per update for 100 updates, and evaluation on DrawBench 200×5 at guidance 1. The development budget uses 16×8 images per update for 50 updates and is scored on DrawBench prompts 1 to 64 with two seeds. Every MEND run trains without guidance except Protocol F, which trains and samples at 4.5. The settings behind each main-text figure and table are:

- Figure 1(a) to (c), Figure 8, Table 3 and Section 4.3: Protocol O, guidance 1, update 100, no checkpoint selection. Panel (a) is the training reward as a trailing mean over five updates.
- Table 2: MEND at guidance 4.5 and update 100; the base and the Flow-GRPO adapter at 4.5; the DiffusionNFT checkpoint at 1. The three-reward row (PickScore/26 + CLIPScore + HPSv2.1) is sampled at guidance 1 at update 300.
- Figure 6: panel (a) at guidance 1 on val64, panels (b) and (c) at 4.5; released checkpoints at their released settings.
- Tables 8 and 11: guidance 1, update 100; Z-Image-Turbo uses 9 Euler steps at 1024 pixels.
- Figure 9 and Table 12: development budget, guidance 1, update 50. Table 13: Protocol O budget, guidance 1, update 50, DrawBench prompts 65 to 200 with two seeds.
- Images: Figures 3, 10 and 11 and columns 1, 4 and 5 of Figure 1 are 1024-pixel renders at each model's headline setting; Figures 5 and 7 and columns 2 and 3 of Figure 1 are 512-pixel renders at guidance 4.5. Figure 4 is logged at update 25 from the rollout policy.
- Appendix figures: in Figure 16, thick curves sample MEND at 4.5 and light curves at 1; reported baselines and our DiffusionNFT run use 1.

B.1 BASELINES

Rows marked † are copied from Zhou et al. (2026); the other rows are re-scored by our harness from released checkpoints, and we train no baseline for a table. Our harness reproduces the reported numbers: SD3.5-M gives PickScore 20.58 against 20.51 at guidance 1 and 22.35 against 22.34 at 4.5, the Flow-GRPO adapter 23.52 against 23.53, and the DiffusionNFT checkpoint 23.82 against 23.80.

The released Flow-GRPO adapter trains for about 4k updates with a KL penalty at guidance 4.5, and the released DiffusionNFT checkpoint trains on five rewards for 1.7k iterations; neither matches

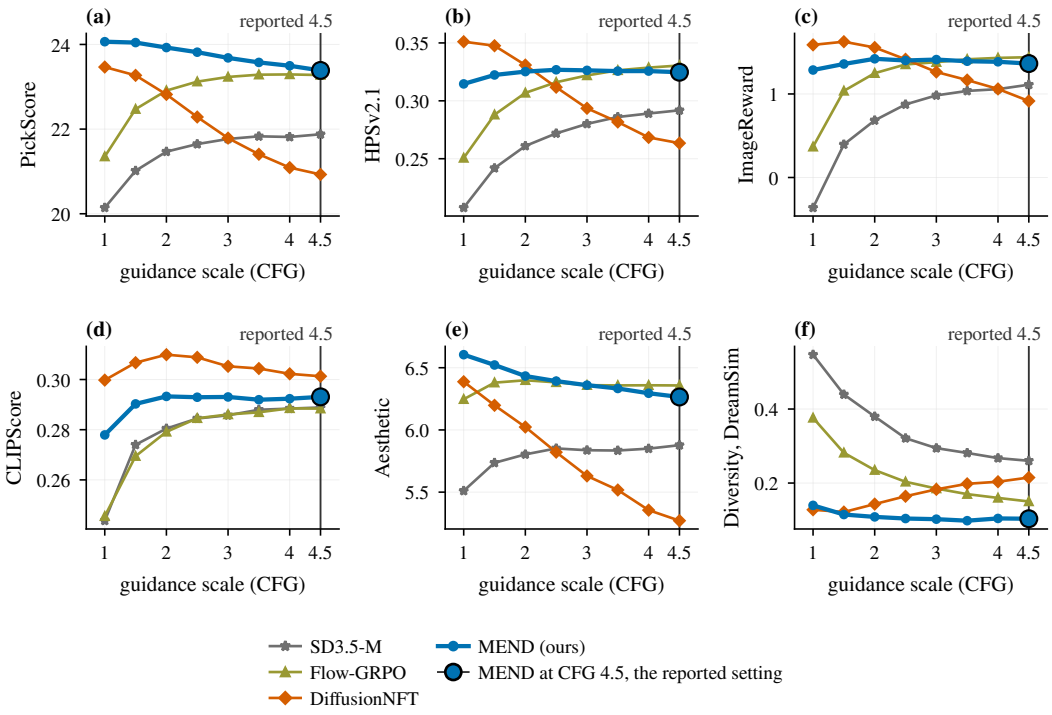

Figure 12: **Guidance trades reward components against diversity.** Val64 (64×2, 512 pixels) for released baselines and MEND at update 100; the marked setting is guidance 4.5.

Protocol O, so the reported rows are the matched comparison. Our own DiffusionNFT run under Protocol O reaches 23.45 against the reported 23.43 and appears only in figures.

B.2 Evaluation protocol

DrawBench (Saharia et al., 2022), 200×5 images, 40-step Euler flow ODE, 512 pixels, EMA weights for MEND. The initial noise of each (prompt, seed) pair is shared across methods, and intervals are 95% prompt-bootstrap intervals. Diversity is the mean pairwise DreamSim (Fu et al., 2023) cosine distance among a prompt's images, averaged over prompts; distance to the base is the DreamSim distance to the base image from the same noise and guidance.

For an image with luma $Y \in [0,1]$, let E_{hf} be the power of the Hann-windowed 2D FFT of $Y - \bar{Y}$ in the band $0.08 \le |f| < 0.25$ cycles per pixel. The HF ratio of a method is $\exp\left(\mathbb{E}_p \mathbb{E}_s \log E_{\mathrm{hf}}(\text{method})/E_{\mathrm{hf}}(\text{ref})\right)$ over prompts p and seeds s, with SD3.5-M at guidance 4.5 as reference.

B.3 Guidance

Figure 12 sweeps guidance on val64, 64 Pick-a-Pic prompts disjoint from DrawBench. MEND's PickScore is highest at guidance 1 and falls slowly, while guidance raises HPSv2.1, ImageReward and CLIPScore and lowers diversity for every model. Protocol F trains MEND at guidance 4.5 for 100 updates without a KL term (Table 4).

Table 4: **Training with guidance.** Protocol F and the released Flow-GRPO PickScore adapter, DrawBench 200×5; both train and sample at guidance 4.5, with different budgets and regularizers.

Method	Upd.	Setting	Reg.	PickScore	HPSv2.1	HPSv3	ImageReward	CLIPScore	Aesthetic	Div.	Dist.
Flow-GRPO, released PickScore adapter	∼4k	4.5/4.5	KL	23.52	0.316	7.04	1.27	0.280	5.90	0.202	0.313
MEND, trained with guidance (Protocol F)	100	4.5/4.5	none	23.56	0.304	5.08	1.07	0.277	5.94	0.214	0.280

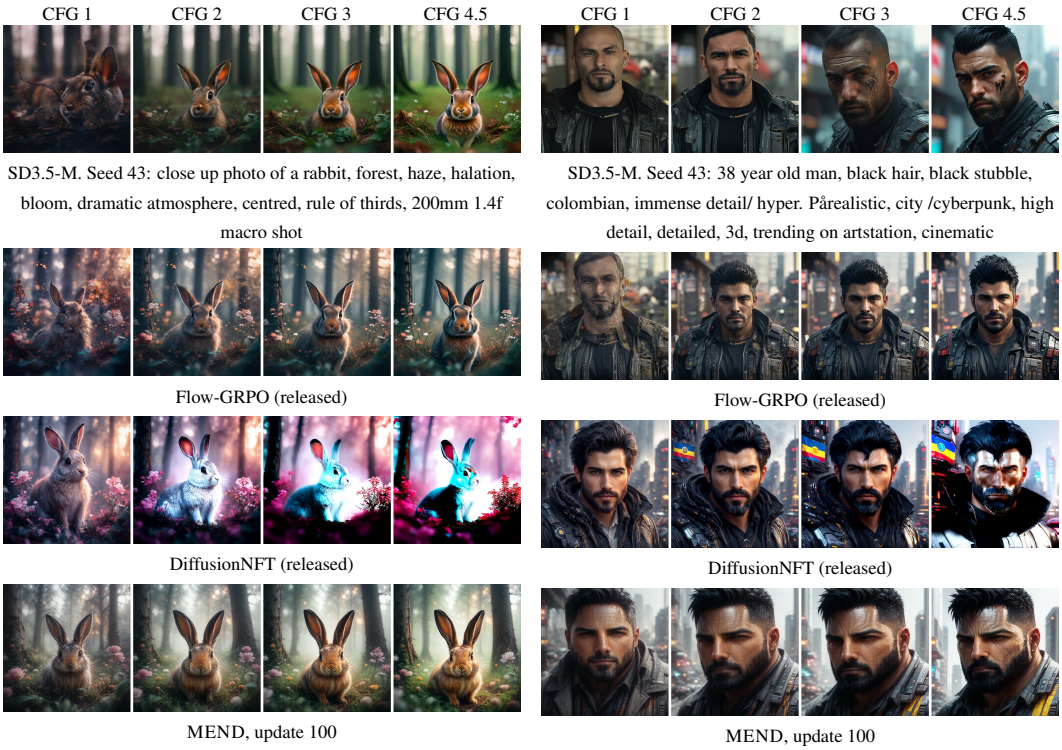

Figure 13: **Guidance sweep in images.** Guidance 1, 2, 3 and 4.5 (columns) for SD3.5-M, released Flow-GRPO, released DiffusionNFT and MEND at update 100 (rows); two val64 prompts, seed 43, 512 pixels, selected by eye. DiffusionNFT over-saturates as guidance grows.

B.4 Other held-out checks

GenEval (Ghosh et al., 2023) and Flow-GRPO's OCR benchmark are run with the scripts of Liu et al. (2025); no MEND run trains on either. At guidance 1 MEND scores 0.62 on GenEval against 0.32 for its base; at guidance 4.5 it matches the Flow-GRPO PickScore adapter on GenEval (0.74 against 0.74) and is below it on OCR (0.53 against 0.68). Figures 14 and 15 place MEND and our DiffusionNFT run on the reward-distance and reward-diversity planes along training.

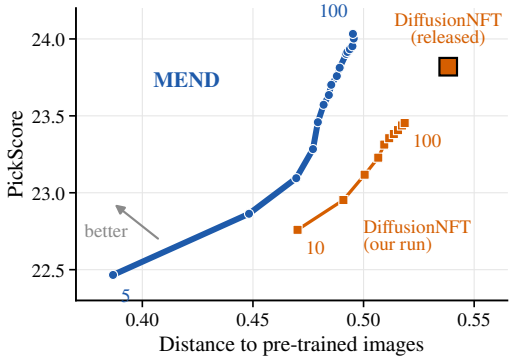

Figure 14: **Higher reward, closer to the pre-trained model.** PickScore against DreamSim distance to the pre-trained images along training (DrawBench 200×5; numbers mark updates). At every update MEND is above our DiffusionNFT run and closer to the pre-trained images.

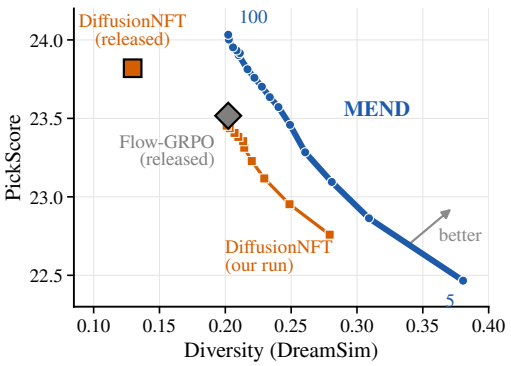

Figure 15: **Higher reward at equal diversity.** PickScore against DreamSim diversity along training (DrawBench 200 × 5; numbers mark updates). At equal diversity MEND is at least 0.35 above our DiffusionNFT run.

C THEORY: STATEMENTS AND PROOFS

This appendix proves the results of Section 3. The guarantees concern target selection and first-order prediction changes, not finite optimizer steps or full training runs (Remark 1).

Conventions. All quantities refer to one round. Latents lie in $\mathbb{R}^D$ with inner product $\langle u, w\rangle = D^{-1}\sum_{j=1}^{D} u_j w_j$ and RMS norm $\|u\| = \langle u, u\rangle^{1/2}$. All latent distances use this norm, with $\mathrm{dist}(x, \emptyset) = +\infty$; W_2^2 denotes the infimum of expected squared distance over couplings. We suppress the prompt and write $R(x) = R_{\mathrm{img}}(\mathrm{Dec}(x), c)$. Its gradient for this inner product is D times the Euclidean gradient. RMS normalization cancels this factor, so the implementation and analysis give the same direction $\hat{h} = g/\|g\|$ for $g \neq 0$.

The behavior adapter rolls out n seeds to endpoints x_i and stored states z_i near t_q. Let $g_i = \nabla R(x_i)$ and $\hat{h}_i = g_i/\|g_i\|$ when $g_i \neq 0$, with $\hat{h}_i = 0$ otherwise. Index the positive proposal lengths so that $\eta_1 \leq \cdots \leq \eta_K$. The accepted set is $\mathcal{A}$, and $d_i = y_i^\star - x_i = \eta_i \hat{h}_i$ for $i \in \mathcal{A}$, where η_i is the selected length; otherwise $d_i = 0$. With $\hat{x}_i^{\mathrm{old}} = \hat{x}_{\theta_{\mathrm{old}}}(z_i)$, equation 4 becomes

$$\mathcal{L}(\theta) = \sum_i w_i \|\hat{x}_\theta(z_i) - \mathrm{sg}[\hat{x}_i^{\mathrm{old}} + d_i]\|^2,$$

$$w_i = \begin{cases} 1/|\mathcal{A}|, & i \in \mathcal{A}, \\ \lambda_{\mathrm{keep}}/(n - |\mathcal{A}|), & i \notin \mathcal{A}. \end{cases}$$

Empty subsets contribute no term, and $\lambda_{\mathrm{keep}} \geq 0$. Parameters lie in $\mathbb{R}^p$ with Euclidean inner product $\langle\cdot,\cdot\rangle_\Theta$ and norm $\|\cdot\|_\Theta$. Write $D_i(\theta) = \partial_\theta \hat{x}_\theta(z_i)$ and define its adjoint by $\langle D_i(\theta)^* u, \phi\rangle_\Theta = \langle u, D_i(\theta)\phi\rangle$. Thus $\nabla_\theta F(\hat{x}_\theta(z_i)) = D_i(\theta)^* \nabla F(\hat{x}_\theta(z_i))$. Operator norms use these inner products, so $\|D_i^*\| = \|D_i\|$, and $u^\top M v = \langle u, Mv\rangle_\Theta$.

The implemented round. Rollouts use θ_{old}, so x_i, z_i and g_i are fixed during regression. For samples below the cap with $g_i \neq 0$, proposals are $y_{ij} = x_i + \eta_j \hat{h}_i$. The verdict maximizes $J_i(y) = R_\kappa(y) - \|y - x_i\|^2/(2\tau)$ over the proposals and x_i. Ties favor x_i, then the shortest proposal. Zero-gradient seeds generate no proposals and are kept. Seeds at or above the cap are also kept: for any $y \neq x_i$, $J_i(y) < J_i(x_i) = \kappa_i$. After the update, $\theta_{\mathrm{old}} \leftarrow \mu_u \theta_{\mathrm{old}} + (1 - \mu_u)\theta$. All statements use exact arithmetic.

Assumptions. Each condition lists the results that require it.

(A1) Frozen round (all results). The behavior parameters, endpoints, stored states, gradients, caps, penalty scale $\tau > 0$ and proposal lengths are fixed before regression and independent of θ. Targets carry no gradient.

(A2) Verdict (all results). Each finite candidate set contains the unchanged sample. Rewards are deterministic, moves require a strict improvement in J_i, and ties follow a fixed rule. Expressions containing $1/|\mathcal{A}|$ assume $\mathcal{A} \neq \emptyset$. If $\mathcal{A} = \emptyset$, $\mathcal{L} = P$, defined below; if $|\mathcal{A}| = n$, the keep term is absent.

(A3) Seeds (Corollaries 3 and 5). A random seed includes its prompt, initial noise and group, which determines its cap. With τ fixed, π and π' are the laws of x and $y^\star$; for a finite round, they are empirical measures. Each gain uses the same seed-specific cap at both endpoints. The relevant maps are measurable, as holds for continuously differentiable rewards and the stated selection rule.

(A4) Smoothness (Propositions 2 and 4, Remark 2). Rewards are differentiable at rollout endpoints, and $\theta \mapsto \hat{x}_\theta(z_i)$ is continuously differentiable. The bound on ∇P additionally assumes $\|D_i(\theta')\| \leq G$ along the segment from θ_{old} to θ. Remark 2 assumes an L_R-Lipschitz reward gradient along the segment from $\hat{x}_i^{\mathrm{old}}$ to x_i.

(A5) Transfer model (Proposition 4 only). The step starts at θ_{old} and uses a symmetric positive semidefinite preconditioner M, fixed conditional on $\mathcal{A}$ and the selected lengths. Gradients satisfy the shared-mean and conditional-independence assumptions stated below. These assumptions are not established for the implemented optimizer; AdamW and gradient clipping are not covered.

C.1 VERIFIED TARGET SELECTION

Proposition 1, full statement (Verified target selection). *(i) Assume (A1) and (A2), and fix a seed with cap κ.*

(ii) Its capped reward gain bounds its quadratic displacement cost:

$$R_\kappa(y^\star) - R_\kappa(x) \geq \frac{\|y^\star - x\|^2}{2\tau} \geq 0.$$

The first inequality is strict whenever $y^\star \neq x$.

(iii) The displacement is bounded by both the longest proposal and the remaining reward below the cap:

$$\|y^\star - x\| \leq \min\left\{\eta_K, \sqrt{2\tau(\kappa - R_\kappa(x))}\right\}.$$

(iv) The raw reward satisfies the same lower bound:

$$R(y^\star) - R(x) \geq \frac{\|y^\star - x\|^2}{2\tau}.$$

Write $r(x) = \sqrt{2\tau(\kappa - R_\kappa(x))}$, suppressing its dependence on the seed's cap.

Corollary 3 (Averaged form). *(i) Assume (A1) to (A3), with π and π' the laws of x and $y^\star$.*

(ii) The mean capped gain bounds the mean squared displacement and the quadratic transport cost:

$$\mathbb{E}[R_\kappa(y^\star) - R_\kappa(x)] \geq \frac{1}{2\tau}\mathbb{E}\|y^\star - x\|^2 \geq \frac{1}{2\tau}W_2^2(\pi, \pi').$$

Proof of Proposition 1. Since x is a candidate, $J(y^\star) \geq J(x) = R_\kappa(x)$, strictly for a move. Rearranging gives

$$\frac{\|y^\star - x\|^2}{2\tau} \leq R_\kappa(y^\star) - R_\kappa(x) \leq \kappa - R_\kappa(x).$$

Thus $\|y^\star - x\| \leq r(x)$. Every proposal has length $\eta_j \leq \eta_K$, giving the other bound. For raw reward, samples at or above the cap are kept and both sides vanish. Otherwise $R_\kappa(x) = R(x)$ and $R(y^\star) \geq R_\kappa(y^\star)$, which gives (iv). □

Proof of Corollary 3. The capped gain is nonnegative, and $\|y^\star - x\|^2 \leq \eta_K^2$. Taking expectations in Proposition 1 gives the first inequality, allowing an infinite expected gain. The joint law of $(x, y^\star)$ couples π and π', so its cost bounds the infimum over all couplings. For a finite round, this coupling is $n^{-1}\sum_i \delta_{(x_i, y_i^\star)}$. □

Remark 1 (The update is not proximal). Proposition 1 concerns targets, not a proximal parameter update. An AdamW step need not realize their displacements or respect their bounds, and shared parameters can change kept samples as well as moved ones (Appendix F.1).

Scope. The guarantees apply to selected targets under the training reward, not to held-out evaluators or the trained model. For a fixed prompt and cap, allowing every latent as a candidate gives the proximal problem $\arg\max_y\{R_\kappa(y) - \|y - x\|^2/(2\tau)\}$, the pointwise problem associated with a JKO step for the potential energy $-\mathbb{E}R_\kappa$ (Jordan et al., 1998). The finite candidate rule never does worse than staying put; no accuracy guarantee relative to the unrestricted optimum is claimed.

C.2 FIRST-ORDER FORM OF THE UPDATE

Proposition 2, full statement (First-order realization). *(i) Assume (A1), (A2) and (A4). Define $\ell_i(\theta) = \langle \hat{h}_i, \hat{x}_\theta(z_i)\rangle$, $e_i(\theta) = \hat{x}_\theta(z_i) - \hat{x}_i^{\text{old}}$ and $P(\theta) = \sum_i w_i\|e_i(\theta)\|^2$.*

(ii) For every θ, the loss separates into a proximity term and the accepted seeds' hint projections:

$$\mathcal{L}(\theta) = P(\theta) - \frac{2}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\eta_i\ell_i(\theta) + C, \qquad C = \sum_i w_i\left(2\langle \hat{x}_i^{\text{old}}, d_i\rangle + \|d_i\|^2\right).$$

Its negative gradient is

$$-\nabla\mathcal{L}(\theta) = \frac{2}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\frac{\eta_i}{\|g_i\|}\nabla_\theta\langle g_i, \hat{x}_\theta(z_i)\rangle - \nabla P(\theta),$$
$$\nabla P(\theta) = 2\sum_i w_i D_i(\theta)^* e_i(\theta).$$

(iii) The proximity gradient vanishes when all $e_i(\theta) = 0$, in particular at θ_{old}. Under the Jacobian bound in (A4),

$$\|\nabla P(\theta)\|_\Theta \le 2(1+\lambda_{\mathrm{keep}})G^2\|\theta - \theta_{\mathrm{old}}\|_\Theta.$$

Proposition 4 (Transfer of the step to held-out seeds)**.** (i) Assume (A1), (A2), (A4) and (A5), with $\Delta\theta = -\alpha M\nabla\mathcal{L}(\theta_{\mathrm{old}})$ and $\alpha > 0$. Conditional on $\mathcal{A}$ and the selected lengths, suppose the gradients $q_s = \nabla\ell_s(\theta_{\mathrm{old}})$ of accepted seeds and a held-out seed $o \notin \mathcal{A}$ satisfy

$$q_s = m + \xi_s, \qquad \mathbb{E}\xi_s = 0, \qquad \mathbb{E}\|\xi_s\|_\Theta^2 \le \sigma^2.$$

Here m and M are fixed, the residuals are independent, and accepted and held-out seeds share the same conditional mean m. The held-out seed is also rolled out by θ_{old}. Write $\bar\eta = |\mathcal{A}|^{-1}\sum_{i\in\mathcal{A}}\eta_i$; all expectations below use this conditioning.

(ii) The expected first-order change in the held-out hint projection depends only on the shared component:

$$\mathbb{E}\langle q_o, \Delta\theta\rangle_\Theta = 2\alpha\bar\eta\, m^\top M m \ge 0.$$

(iii) The step consists of a shared component and a zero-mean residual:

$$\Delta\theta = 2\alpha\bar\eta M m + s, \qquad s = \frac{2\alpha}{|\mathcal{A}|}M\sum_{i\in\mathcal{A}}\eta_i\xi_i,$$
$$\mathbb{E}\|s\|_\Theta^2 \le \frac{4\alpha^2\|M\|^2\eta_K^2\sigma^2}{|\mathcal{A}|}.$$

(iv) For each accepted seed $k \in \mathcal{A}$, its expected first-order change exceeds the held-out value by at most

$$0 \le \mathbb{E}\langle q_k, \Delta\theta\rangle_\Theta - \mathbb{E}\langle q_o, \Delta\theta\rangle_\Theta \le \frac{2\alpha\eta_K\|M\|\sigma^2}{|\mathcal{A}|}.$$

Since $\langle q_s, \Delta\theta\rangle_\Theta = \langle \hat{h}_s, D_s(\theta_{\mathrm{old}})\Delta\theta\rangle$, Proposition 4 concerns the linearized projection of a clean prediction onto its endpoint hint, not the change in endpoint reward or a held-out evaluator.

Proof of the first-order realization result. By (A1), $\hat{x}_i^{\mathrm{old}}$ and d_i are constant with respect to θ. Expanding each squared error gives

$$\|e_i(\theta) - d_i\|^2 = \|e_i(\theta)\|^2 - 2\langle\hat{x}_\theta(z_i), d_i\rangle + 2\langle\hat{x}_i^{\mathrm{old}}, d_i\rangle + \|d_i\|^2.$$

Sum with weights w_i and substitute $d_i = \eta_i\hat{h}_i$ on $\mathcal{A}$ and $d_i = 0$ elsewhere to obtain the loss identity. Differentiating gives (ii), since $\hat{h}_i = g_i/\|g_i\|$ on $\mathcal{A}$ and the derivative of $\|e_i(\theta)\|^2$ is $2D_i(\theta)^* e_i(\theta)$. This derivative vanishes at θ_{old}.

For (iii), the mean value inequality gives $\|e_i(\theta)\| \le G\|\theta - \theta_{\mathrm{old}}\|_\Theta$. Since $\|D_i(\theta)^*\| \le G$,

$$\|\nabla P(\theta)\|_\Theta \le 2G^2\|\theta - \theta_{\mathrm{old}}\|_\Theta\sum_i w_i \le 2(1+\lambda_{\mathrm{keep}})G^2\|\theta - \theta_{\mathrm{old}}\|_\Theta.$$

The weight sum is $1 + \lambda_{\mathrm{keep}}$ when both subsets are nonempty, 1 when all seeds move, and λ_{keep} when none move. □

Proof of Proposition 4. At θ_{old}, the proximity gradient vanishes, giving

$$\Delta\theta = \frac{2\alpha}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\eta_i M q_i.$$

All quantities below are conditional on $\mathcal{A}$ and the selected lengths. Finite second moments make the products integrable. Independence and zero means give

$$\mathbb{E}\langle m, M\xi_s\rangle_\Theta = 0, \qquad \mathbb{E}\langle \xi_s, M\xi_{s'}\rangle_\Theta = 0 \quad (s \neq s'),$$

while positive semidefiniteness gives $0 \le \xi_s^\top M\xi_s \le \|M\|\|\xi_s\|_\Theta^2$.

For (ii), the held-out residual is independent of every accepted residual, so $\mathbb{E}\langle q_o, Mq_i\rangle_\Theta = m^\top Mm$. Summing yields $\mathbb{E}\langle q_o, \Delta\theta\rangle_\Theta = 2\alpha\bar{\eta}\, m^\top Mm \ge 0$.

For (iii), substitute $q_i = m + \xi_i$ into the step. The residual s has zero mean, and its cross terms vanish, giving

$$\mathbb{E}\|s\|_\Theta^2 \le \frac{4\alpha^2\|M\|^2}{|\mathcal{A}|^2}\sum_{i\in\mathcal{A}}\eta_i^2\mathbb{E}\|\xi_i\|_\Theta^2 \le \frac{4\alpha^2\|M\|^2\eta_K^2\sigma^2}{|\mathcal{A}|}.$$

For (iv), only the self-term $i = k$ differs from the held-out calculation:

$$\mathbb{E}\langle q_k, \Delta\theta\rangle_\Theta = 2\alpha\bar{\eta}\, m^\top Mm + \frac{2\alpha\eta_k}{|\mathcal{A}|}\mathbb{E}[\xi_k^\top M\xi_k].$$

The final term lies between 0 and $2\alpha\eta_K\|M\|\sigma^2/|\mathcal{A}|$. □

Remark 2 (Relation to one-step reward backpropagation). Let $D_i = D_i(\theta_{\text{old}})$. The accepted-seed term uses $D_i^*\nabla R(x_i)$, whereas one-step reward backpropagation (Xu et al., 2023; Clark et al., 2024) uses $D_i^*\nabla R(\hat{x}_i^{\text{old}})$. Under the Lipschitz condition in (A4),

$$\|D_i^*\big(\nabla R(x_i) - \nabla R(\hat{x}_i^{\text{old}})\big)\|_\Theta \le \|D_i\|L_R\|x_i - \hat{x}_i^{\text{old}}\|.$$

Multiplying by $2\eta_i/(|\mathcal{A}|\|g_i\|)$ gives the discrepancy between the corresponding weighted gradient terms. The bound is qualitative: the endpoint-to-prediction gap is not assumed small.

The EMA lag. The implemented step starts at θ, not θ_{old}, so its loss gradient includes the proximity term. Let $\Gamma_u = \theta - \theta_{\text{old}}$ after update u and the behavior EMA, and let $\Delta\theta_u$ be that update. The recursion $\Gamma_u = \mu_u(\Gamma_{u-1} + \Delta\theta_u)$ gives

$$\|\Gamma_u\|_\Theta \le \Big(\prod_{v=1}^{u}\mu_v\Big)\|\Gamma_0\|_\Theta + \frac{\bar{\mu}}{1-\bar{\mu}}\max_{1\le v\le u}\|\Delta\theta_v\|_\Theta$$

whenever $0 \le \mu_v \le \bar{\mu} < 1$. With zero initial lag and $\bar{\mu} = 0.1$, the lag is at most one ninth of the largest step over the updates considered. Proposition 2(iii) bounds the proximity gradient by $2(1+\lambda_{\text{keep}})G^2\|\Gamma_u\|_\Theta$. This term anchors predictions to the behavior adapter, not the base model.

Interpreting the update. At θ_{old}, the descent direction uses reward gradients evaluated at rollout endpoints, normalized in latent space, and filtered and scaled by the verdict. For the idealized step of Proposition 4, writing $D_s = D_s(\theta_{\text{old}})$ gives

$$D_s\Delta\theta = \frac{2\alpha}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} D_sMD_i^*d_i.$$

Thus the linearized prediction change is a kernel-weighted combination of accepted displacements. Proposition 4 describes its expected transfer under (A5).

C.3 Target mass across region boundaries

Corollary 5 (Mode mass of the targets). *(i) Assume (A1) to (A3). Let $\rho(x) = \min\{\eta_K, r(x)\}$ and fix a Borel set $A \subseteq \mathbb{R}^D$.*

(ii) The change in probability assigned to A satisfies

$$|\pi'(A) - \pi(A)| \le \Pr\big[\operatorname{dist}(x, \partial A) \le \|y^\star - x\|\big] \le \Pr\big[\operatorname{dist}(x, \partial A) \le \rho(x)\big].$$

(iii) A region can receive targets only from samples within their displacement bound of it:

$$\pi'(A) \le \Pr\big[\operatorname{dist}(x, A) \le \rho(x)\big].$$

Proof. The distance functions are 1-Lipschitz for nonempty sets and identically $+\infty$ otherwise, so the events are measurable by (A3). Under the seed coupling,

$$\pi'(A) - \pi(A) = \Pr[y^\star \in A, x \notin A] - \Pr[x \in A, y^\star \notin A].$$

Its absolute value is at most the probability that exactly one endpoint belongs to A. In that event, the segment from x to $y^\star$ meets ∂A: otherwise the disjoint open sets $\operatorname{int} A$ and $\operatorname{int}(A^c)$ would disconnect the segment. Hence $\operatorname{dist}(x, \partial A) \le \|y^\star - x\| \le \rho(x)$, proving (ii). Finally, $y^\star \in A$ implies $\operatorname{dist}(x, A) \le \|y^\star - x\| \le \rho(x)$, proving (iii). □

Samples at the cap have $\rho(x) = 0$, and regions outside every sample's displacement bound receive no target mass. In contrast, consider the KL tilt $\pi_\beta(dx) \propto e^{R(x)/\beta}\pi(dx)$, with $\beta > 0$ and finite normalizer. For $\pi(A) > 0$,

$$\frac{\pi_\beta(A)}{\pi(A)} = \frac{\mathbb{E}_\pi[e^{R/\beta} \mid A]}{\mathbb{E}_\pi[e^{R/\beta}]}.$$

If $R \le r_A$ on A and $R \ge r_A + \Delta$ off A, then

$$\frac{\pi_\beta(A)}{\pi(A)} \le \frac{1}{\pi(A) + (1 - \pi(A))e^{\Delta/\beta}},$$

regardless of the distance between A and the remaining support. These are target-level statements; diversity after training is evaluated empirically.

D Full result tables

Table 5 extends Table 2 with every evaluator, diversity, distance to the base and 95% intervals.

Table 5: **Full SD3.5-M results.** DrawBench 200 × 5; setting is training/evaluation guidance, † marks reported rows (n/r: not reported) and * the default Adam ϵ. Grey sub-rows give 95% prompt-bootstrap intervals; **bold** and underline mark the best and second best trained headline rows.

Method	Upd.	Setting	Reg.	PickScore	HPSv2.1	HPSv3	ImageReward	CLIPScore	Aesthetic	Div. ↑	Dist. ↓
Headline settings (Appendix B)											
SD3.5-M	0	–/4.5		22.35	0.280	2.72	0.83	0.283	5.39	0.325	0
95% interval				[22.15, 22.54]	[0.274, 0.286]	[2.13, 3.27]	[0.72, 0.95]	[0.277, 0.290]	[5.31, 5.47]	[0.307, 0.344]	n/a
Flow-GRPO, released PickScore adapter	∼4k	4.5/4.5	KL	23.52	0.316	7.04	1.27	0.280	5.90	**0.202**	**0.313**
95% interval				[23.29, 23.73]	[0.310, 0.321]	[6.63, 7.42]	[1.17, 1.36]	[0.273, 0.286]	[5.82, 5.97]	[0.189, 0.216]	[0.298, 0.329]
DiffusionNFT, released multi-reward	1.7k	1/1	MSE	**23.82**	**0.331**	**7.50**	**1.49**	**0.292**	**6.02**	0.130	0.538
95% interval				[23.61, 24.02]	[0.326, 0.336]	[7.13, 7.85]	[1.41, 1.56]	[0.286, 0.298]	[5.94, 6.09]	[0.123, 0.137]	[0.523, 0.554]
MEND (ours)	100	1/4.5	none	23.70	0.319	7.15	1.32	0.291	5.88	0.144	**0.313**
95% interval				[23.48, 23.91]	[0.314, 0.324]	[6.71, 7.57]	[1.23, 1.41]	[0.284, 0.297]	[5.80, 5.97]	[0.133, 0.156]	[0.297, 0.329]
Matched 100-update comparison (Protocol O)											
SD3.5-M	0	–/1		20.58	0.207	−6.44	−0.52	0.239	5.14	0.578	0
95% interval				[20.44, 20.71]	[0.203, 0.211]	[−6.87, −5.99]	[−0.63, −0.42]	[0.233, 0.245]	[5.09, 5.20]	[0.564, 0.592]	n/a
MEND	100	1/1	none	24.03	0.301	3.71	1.13	0.273	6.24	0.202	0.495
95% interval				[23.81, 24.24]	[0.296, 0.306]	[3.24, 4.17]	[1.03, 1.23]	[0.266, 0.279]	[6.17, 6.31]	[0.188, 0.218]	[0.481, 0.510]
MEND, default Adam ϵ^*	100	1/1	none	23.90	0.298	3.75	1.07	0.271	6.18	0.212	0.491
ReFL†	100	1/1	none	23.92	n/r	n/r	n/r	n/r	n/r	n/r	n/r
DiffusionNFT†	100	1/1	MSE	23.43	n/r	n/r	n/r	n/r	n/r	n/r	n/r

D.1 Validation split

Every design decision was made at the development budget on DrawBench prompts 1 to 64. Table 6 repeats Table 2 on DrawBench prompts 65 to 200, which no decision has seen.

E Training dynamics and transfer

Figure 16 follows the per-reward runs of Section 4.4 through training.

Table 6: **Held-out DrawBench prompts 65 to 200.** Rows at guidance 4.5 reuse the 136×5 images of Table 2; other rows use 136×2 images. HF is a ratio to SD3.5-M at guidance 4.5 and is not ranked.

Method	Setting	PickScore	HPSv2.1	HPSv3	ImageReward	CLIPScore	Aesthetic	Div.	HF
SD3.5-M	1	20.62	0.209	−6.58	−0.49	0.239	5.16	**0.568**	0.40
SD3.5-M	4.5	22.39	0.283	2.85	0.87	0.284	5.43	0.321	1.00
Flow-GRPO, released PickScore adapter	4.5	23.52	0.317	7.03	1.31	0.281	5.95	0.201	0.96
DiffusionNFT, released multi-reward	1	23.81	**0.332**	**7.37**	**1.53**	**0.293**	6.04	0.128	0.56
MEND	4.5	23.75	0.322	7.30	1.34	0.291	5.95	0.143	1.34
MEND	1	**24.12**	0.304	3.82	1.21	0.274	**6.31**	0.184	0.52
MEND, default Adam ϵ	1	23.98	0.301	3.90	1.14	0.274	6.22	0.195	0.48

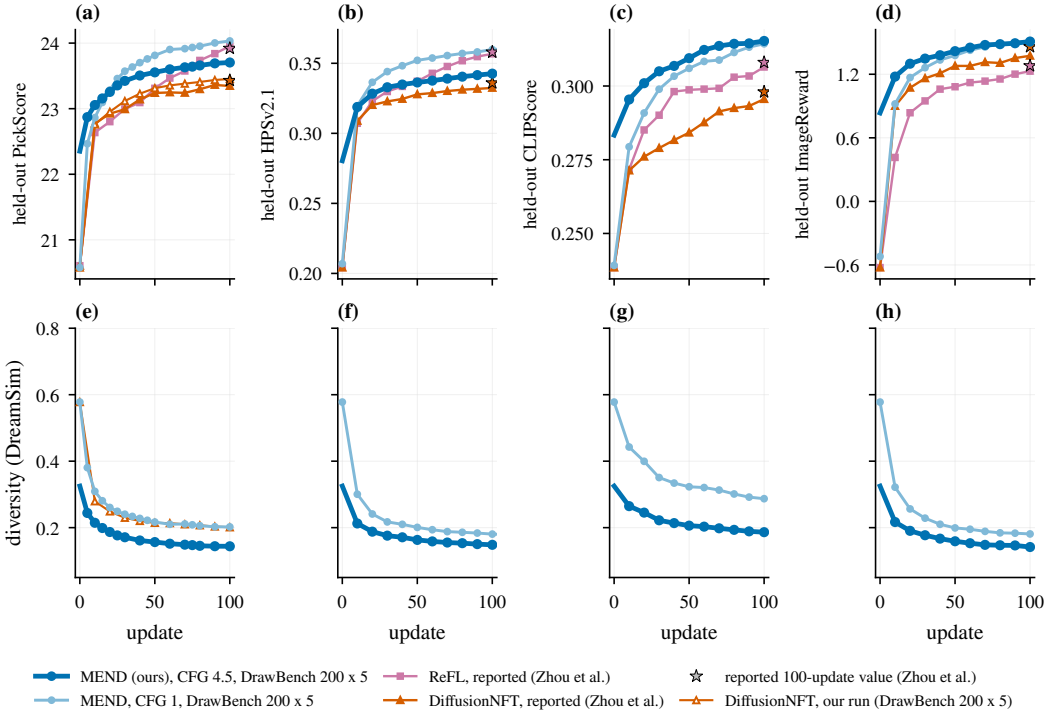

Figure 16: **Held-out reward and diversity for each training reward.** Single-seed MEND runs on DrawBench 200×5; reported ReFL and DiffusionNFT curves are digitized from Zhou et al. (2026, Figure 8), and open triangles mark our DiffusionNFT run.

E.1 Other rewards and SD3-M

Table 7 holds one run per training reward and Table 8 the run on SD3-M. The ‡ row gives, for each run, the last update before a held-out evaluator on val64 falls below its best earlier value by more than its 95% paired bootstrap half-width, a rule fixed before the numbers were read.

Table 7: **One run per training reward.** SD3.5-M, Protocol O, update 100, DrawBench 200×5; †: reported by Zhou et al. (2026), *: default Adam ϵ, ‡: last update before held-out decline. The HPSv3 run is in Table 9.

Method	PickScore	CLIPScore	HPSv2.1	HPSv3	ImageReward
SD3.5-M†	20.51	0.237	0.204	−6.47	−0.58
SD3.5-M, measured by us	20.58	0.239	0.207	−6.44	−0.52
ReFL†	23.92	0.308	0.358	9.33	1.28
DiffusionNFT†	23.43	0.298	0.336	9.14	1.46
MEND	24.03	0.314	0.360	–	1.51
MEND, default ϵ*	23.90	0.306	0.358	–	1.44
MEND, before decline‡	–	0.306	0.358	–	1.17

Table 8: **SD3-M transfer.** DrawBench 200 × 5, Protocol O, update 100; released Linear-DPO (Li et al., 2026) is re-scored.

Model	PickScore	HPSv2.1	CLIPScore	Div. ↑	Dist. ↓
SD3-M	20.36	0.215	0.234	0.552	0
Linear-DPO, released	20.96	0.256	0.241	0.464	0.341
MEND, PickScore	23.70	0.298	0.263	0.239	0.500

E.2 Every evaluator per run

Table 9 scores each single-reward run with every evaluator, including a fifth run on HPSv3 whose last full evaluation is update 90.

Table 9: **Every evaluator for each single-reward run.** DrawBench 200 × 5; the training reward of each row is its diagonal entry. Top block at guidance 1, bottom block at guidance 4.5.

Training reward	PickScore	HPSv2.1	HPSv3	ImageReward	CLIPScore	Aesthetic	Div. ↑	Dist. ↓
none (SD3.5-M)	20.58	0.207	−6.44	−0.52	0.239	5.14	0.578	0
PickScore (seed 1)	24.03	0.301	3.71	1.13	0.273	6.24	0.202	0.495
CLIPScore	22.27	0.268	1.15	0.98	0.314	5.22	0.287	0.450
HPSv2.1	23.06	0.360	7.12	1.21	0.268	5.89	0.181	0.519
ImageReward	22.33	0.303	4.78	1.51	0.266	5.99	0.181	0.535
HPSv3 (update 90)	22.28	0.304	8.75	0.83	0.266	5.95	0.310	0.467
Headline evaluation settings (Appendix B)								
PickScore (seed 1)	23.70	0.319	7.15	1.32	0.291	5.88	0.144	0.313
CLIPScore	22.58	0.298	5.44	1.27	0.315	5.40	0.186	0.303
HPSv2.1	22.94	0.343	7.37	1.29	0.285	5.61	0.148	0.328
ImageReward	22.41	0.311	5.53	1.51	0.284	5.61	0.142	0.354

E.3 Training past the budget and over-optimization

Trained on PickScore for 200 updates (Table 10, top; Figure 17), MEND keeps raising the trained reward while held-out gains flatten and diversity declines slowly. Trained on the LAION aesthetic predictor, a reward known to be over-optimized (Black et al., 2024; Clark et al., 2024), it raises the aesthetic score from 5.2 to 9.7 while every other evaluator collapses (Table 10, bottom).

The verdict certifies against the training reward only, so it cannot prevent this. Continuing the HPSv2.1 run from update 100 to 200 likewise raises HPSv2.1 while held-out HPSv3 falls, consistent with the ‡ row of Table 7.

Table 10: **200 updates at the development budget.** DrawBench prompts 65 to 200 × 2 seeds. Top: PickScore training; bottom: LAION aesthetic training.

Training reward	Updates	PickScore	HPSv3	ImageReward	CLIPScore	Aesthetic	Div.	Dist.
none (SD3.5-M)	0	20.62	−6.58	−0.49	0.239	5.16	0.568	0
PickScore	50	23.15	2.93	0.81	0.262	6.15	0.271	0.457
	100	23.47	3.02	0.95	0.265	6.21	0.252	0.469
	150	23.65	3.31	0.99	0.266	6.28	0.240	0.476
	200	23.78	3.34	1.04	0.270	6.22	0.238	0.474
Aesthetic	50	19.90	−5.90	−0.84	0.155	8.84	0.200	0.634
	100	18.91	−6.28	−1.27	0.123	9.35	0.198	0.677
	150	18.58	−6.76	−1.48	0.107	9.53	0.185	0.695
	200	18.43	−6.71	−1.56	0.099	9.66	0.192	0.703

E.4 Z-Image-Turbo

Z-Image-Turbo (Z-Image Team et al., 2025) is a distilled model sampled in 9 steps at 1024 pixels; we follow Zhou et al. (2026, Appendix C.1) with 1024-pixel rollouts and 48 × 12 images per update. Both runs improve every held-out evaluator except CLIPScore (Table 11).

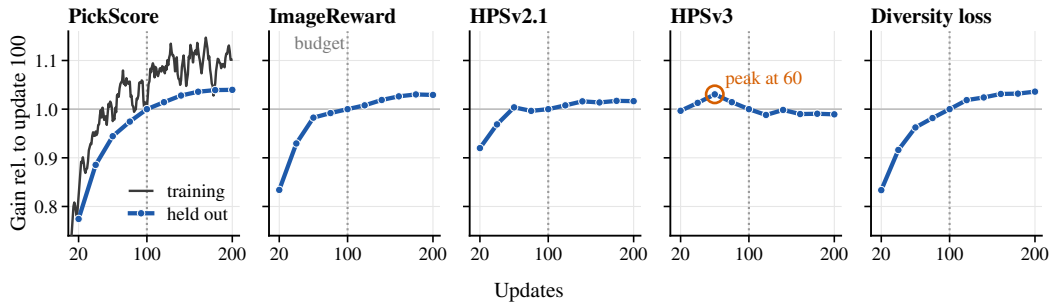

Figure 17: **Training past the budget.** A second PickScore run trained for 200 updates; each panel shows the change from the base relative to the change at update 100 (dotted line). The trained reward keeps rising while held-out gains flatten and HPSv3 peaks at update 60.

Table 11: **Z-Image-Turbo, every evaluator.** DrawBench 200 × 5, 1024 pixels, 9 steps.

Model	PickScore	HPSv2.1	HPSv3	ImageReward	CLIPScore	Aesthetic	Div. ↑	Dist. ↓
Z-Image-Turbo	22.86	0.295	6.27	0.97	0.275	5.42	0.155	0
MEND, trained on PickScore	24.10	0.315	7.85	1.22	0.271	6.06	0.116	0.199
MEND, trained on HPSv2.1	23.27	0.357	9.53	1.20	0.275	5.86	0.120	0.172

F Ablations and controls

Table 12 varies one part of MEND at a time at the development budget; "fixed length" rescales every trained move to RMS 0.1, and "self-normalized loss" divides each seed's error by the squared length of its move. A looser cap, a price band with more acceptances and a higher learning rate all trade diversity for reward.

Table 12: **Development budget ablations.** SD3.5-M, 16 × 8 images per update, 50 updates, one training seed per arm; last two columns on DrawBench 200 × 5 where measured. Moved: mean share of seeds receiving a move target.

			DrawBench 1 to 64 × 2 (development)				DrawBench 200 × 5	
	Arm	Moved	PickScore	HPSv3	ImageReward	Div.	PickScore	Div.
	SD3.5-M, no update	n/a	20.52	−6.77	−0.58	0.589	20.58	0.578
	MEND (explicit candidates; verdict, price, cap; single state; certified move)	0.34	22.99	3.03	0.66	0.318	23.10	0.297
	MEND, $\epsilon = 10^{-12}$	0.34	23.10	3.12	0.76	0.292	–	–
candidates	One candidate ($K = 1, \eta = 0.1$)	0.37	23.03	3.33	0.71	0.302	23.20	0.280
candidates	Two states in $[0.2, 0.6]$, $\eta \in \{0.05, 0.1, 0.2\}$	0.37	22.88	3.16	0.68	0.335	–	–
candidates	Anchored restart proposals from $t = 0.86$	0.41	22.79	2.48	0.67	0.316	–	–
candidates	Two gradient steps per proposal	0.36	22.98	2.01	0.65	0.292	–	–
candidates	Reward gradient clipped to unit norm	0.35	23.18	3.17	0.73	0.293	–	–
verdict, cap	No price ($\tau = \infty$): best capped reward wins	0.78	23.05	2.96	0.67	0.310	23.21	0.291
verdict, cap	No cap ($\kappa = \infty$), verdict on	0.45	23.17	3.24	0.74	0.304	23.27	0.287
verdict, cap	Price band $[0.6, 0.8]$	0.54	23.09	3.32	0.72	0.297	–	–
verdict, cap	Cap quantile $q = 0.9$	0.42	23.03	3.48	0.72	0.300	–	–
verdict, cap	No per-seed verdict: every seed below the cap takes $\eta = 0.2$	0.81	23.35	3.60	0.81	0.264	23.45	0.260
verdict, cap	No per-seed verdict: every seed below the cap takes $\eta = 0.1$	0.81	23.00	2.92	0.66	0.316	23.16	0.301
verdict, cap	Endpoint check only: one $\eta = 0.2$ candidate kept iff $R(y) > R(x)$; no price, no cap	0.83	23.22	3.52	0.76	0.299	23.34	0.280
target	No verdict, no cap, fixed length 0.1 on every seed, self-normalized loss	1.00	23.59	3.51	0.95	0.216	–	–
target	same, two states, squared loss	1.00	23.54	5.37	0.91	0.216	–	–
target	+ verdict and cap ($\eta \in \{0.05, 0.1, 0.2\}$)	0.37	23.16	3.27	0.80	0.291	–	–
target	squared loss (fixed-length variant)	0.37	23.16	3.52	0.82	0.299	–	–
target	Two trained states and learning rate 6×10^{-4} (two factors change)	0.36	23.12	3.38	0.76	0.300	–	–
optimizer	Learning rate 4.5×10^{-4}	0.34	23.02	3.12	0.60	0.292	–	–
optimizer	Learning rate 6×10^{-4}	0.34	23.19	2.81	0.83	0.277	–	–
optimizer	with cap quantile $q = 0.5$	0.30	23.10	2.83	0.72	0.274	–	–
optimizer	with cap quantile $q = 0.6$	0.31	23.26	2.83	0.81	0.280	–	–
optimizer	with cap quantile $q = 0.9$	0.39	23.31	3.49	0.83	0.275	–	–
optimizer	Learning rate 10^{-3}	0.34	23.34	3.49	0.88	0.269	–	–

F.1 Design choices

Candidates are explicit moves of the endpoint, decoded and scored as they are; re-noising the endpoint and denoising again (Meng et al., 2022; Bansal et al., 2024) trained less cleanly in development runs. Nearly every accepted move takes the shortest rung $\eta = 0.1$, because the longer moves lose on reward and not only on price, so the longer rungs check that the shortest move lies on the rising part of the ray.

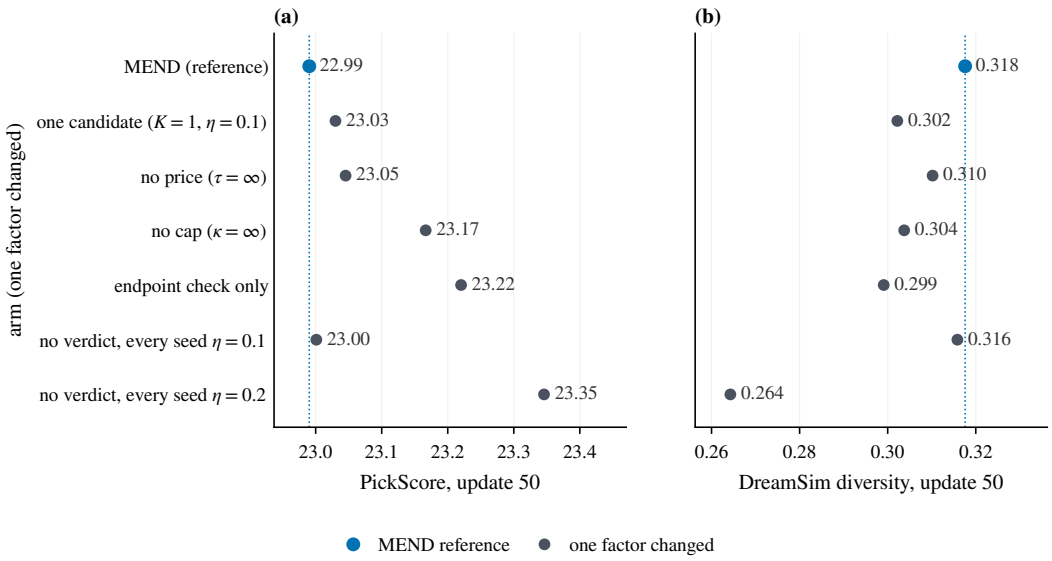

Figure 18: **One-factor arms at the development budget.** PickScore and diversity of the arms of Table 12; dotted lines mark MEND.

The verdict certifies targets; the optimizer decides how far the model moves. In the two Protocol O runs, the mean squared endpoint move of 12 held-out seeds re-rolled after each update has a median of 0.78 and 1.03 times the mean certified squared move. Seeds the verdict would keep move as much as seeds it would move, and the median projection of the move onto the certified direction is 0.001.

F.2 Sensitivity at the full budget

Table 13 varies one factor at the Protocol O budget, stopped at update 50. A second launch of the reference differs by 0.01 PickScore, the initial price and the EMA barely matter, a 0.05 rung lowers PickScore, and adding HPSv2.1 to the reward raises HPSv2.1 and HPSv3 at no PickScore cost.

Table 13: **Sensitivity at the full budget.** Protocol O budget, update 50, one training seed per arm, DrawBench prompts 65 to 200 $\times$ 2 seeds; last column on DrawBench 200 $\times$ 5 where measured. τ_{50}: the price at update 50.

Arm	τ_{50}	PickScore	HPSv2.1	HPSv3	ImageReward	CLIPScore	Aesthetic	Div.↑	Dist.↓	PickScore, 200 × 5
MEND (reference run, $\tau_0 = 0.1$)	0.225	23.88	0.304	4.00	1.15	0.269	6.30	0.208	0.484	23.81
Same configuration, second launch	0.225	23.89	0.303	4.07	1.14	0.270	6.32	0.202	0.486	23.83
$\tau_0 = 0.45$	0.200	23.90	0.302	3.86	1.12	0.270	6.33	0.203	0.487	23.83
$\tau_0 = 0.9$	0.267	23.94	0.303	3.95	1.16	0.270	6.29	0.211	0.484	23.81
One candidate ($K = 1$, $\eta = 0.1$)	0.225	23.90	0.302	3.74	1.14	0.271	6.26	0.203	0.487	23.79
Five candidates ($\eta \in \{0.05, 0.1, 0.2, 0.4, 0.8\}$)	0.067	23.63	0.299	3.63	1.02	0.266	6.25	0.224	0.475	–
No EMA (behavior adapter = trained adapter)	0.225	23.91	0.306	4.17	1.13	0.273	6.30	0.213	0.485	–
PickScore/26 + HPSv2.1, unit weights	0.150	23.85	0.345	6.53	1.25	0.276	6.26	0.177	0.495	–

F.3 One-dimensional toy model

The data density of Figure 2 is a mixture of three Gaussians with means 2.2, 0, -2.2, weights 0.22, 0.43, 0.35 and standard deviations 0.34, 0.17, 0.20; paths are quantile curves of the exact 1D rectified flow from $\mathcal{N}(0, 1)$. The true reward is a bump of width 0.2 at each mode, of height 0.85 for the top mode and 1 otherwise, and the learned reward is $\hat{R} = R + 0.45\,\mathrm{softplus}((|x| - 2.27)/0.1)$. Panel (b) is $p\,e^{R/\beta}$ with $\beta = 0.05$; panel (c) takes 18 steps $x \leftarrow x + 0.03\,\hat{R}'(x)$; panel (d) shows the targets $x + d$ of one MEND round on 40,000 samples with groups of 8, $q = 0.75$, $\eta \in \{0.15, 0.3, 0.6\}$ and $\tau = 0.25$.

After the round the three modes hold 0.218, 0.430 and 0.353 of the mass (0.217, 0.430 and 0.350 before), while the ascent of panel (c) moves 0.127 of the mass beyond $|x| = 3.25$, where the data has none.

F.4 THE TOY OVER REPEATED ROUNDS

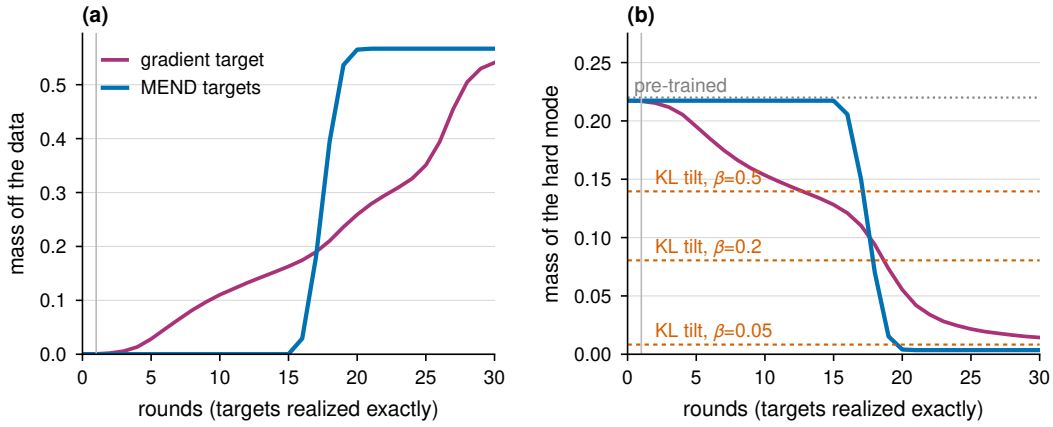

Figure 19: **The cap delays drift under a wrong reward.** The three-mode toy over repeated rounds with targets realized exactly: off-data mass and retained hard-mode mass. Dashed lines: KL-tilt fixed points.

For 15 rounds MEND's targets keep every mode's mass and put at most 0.0003 of the mass off the data, while gradient ascent drains the hard mode and moves 16% of the mass off the data. From round 16 the learned reward, which keeps rising beyond the data, certifies moves off it, and by round 18 MEND has passed gradient ascent.

Proposition 1 certifies gains in the learned reward; when that reward is wrong in a direction the samples can reach, verification cannot detect the error and the cap only postpones it. This is why we report held-out evaluators and diversity over training.

G MORE QUALITATIVE RESULTS

Sample prompts and seeds are selected by eye from images rendered with the same prompts and initial noise for every model; no image was edited, and every number in the tables is over the full prompt sets.

Prompts of Figures 1 to 11, 22, 26 and 27. Figure 1, top part, columns left to right. (1) seed 0: "A lighthouse on a cliff to the left and a sailboat to the right on a calm sea, a full moon between them, blue hour."; (2) seed 45: "Four dogs on the street."; (3) seed 43: "A small blue book sitting on a large red book."; (4) seed 1: "A miniature train set winding through a handmade model town with tiny people, trees and street lamps, tilt-shift photograph."; (5) seed 0: "A steaming bowl of ramen with a soft-boiled egg, sliced pork, scallions and nori, dark wooden table, moody food photography.".

Figure 3, rows top to bottom. (1) seed 0: "A cat wearing a tiny knitted scarf sitting inside a teacup, next to a stack of three books, cozy window light."; (2) seed 1: "A cute ceramic dragon figurine with glossy teal glaze, isometric 3D render, soft pastel background, clay material."; (3) seed 1: "Close-up of a peacock feather showing iridescent blue and green barbules, studio macro, black background."; (4) seed 0: "A giant tortoise carrying a small village with tiny houses and trees on its shell, walking through a shallow sea, fantasy illustration."; (5) seed 42: "multidimensional quantum foam realm brane fantasy style".

Figure 5, columns left to right. (1) seed 42: "An umbrella on top of a spoon."; (2) seed 43: "A tennis racket underneath a traffic light."; (3) seed 46: "Three dogs on the street."; (4) seed 42: "A yellow book and a red vase."; (5) seed 46: "A green apple and a black backpack."; (6) seed 45: "A bird scaring a scarecrow.".

Figure 7, rows top to bottom. (1) seed 45: "A spider with a moustache bidding an equally gentlemanly grasshopper a good day during his walk to work."; (2) seed 43: "A medieval painting of the wifi not working.".

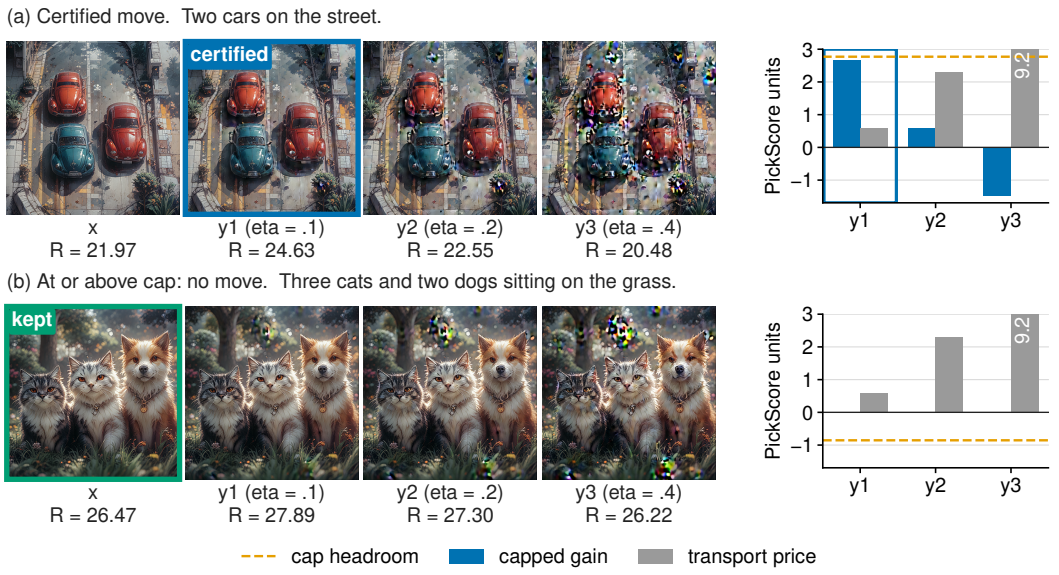

Figure 20: **A certified move and a kept seed.** Two logged seeds of the rollout policy at update 100, each with its three proposals; bars compare capped gain with price and boxes mark the selected target. Across 1,200 logged seeds, 33% were certified, all at the shortest step.

Figure 10, columns left to right. (1) seed 1: "Desert canyon with glowing red sandstone walls at dusk, a thunderstorm on the horizon with a single lightning bolt, wide-angle photograph."; (2) seed 0: "An ancient tree with a glowing door carved into its trunk, fireflies and luminous mushrooms around its roots, enchanted night forest."; (3) seed 43: "A badge, flag , icon, space, space program,"; (4) seed 0: "Ornate sapphire tiara in a dim museum vitrine, blue stones and diamonds sparkling under spotlight with glass reflections."; (5) seed 42: "a treehouse on a palm tree"; (6) seed 1: "A chef's hands dusting flour over fresh handmade pasta on a wooden board, flour particles in the air, side light.".

Figure 11, rows top to bottom. (1) seed 0: "Looking up at a spiral staircase inside an old library with carved wooden railings and shelves of leather-bound books on every floor."; (2) seed 0: "Surreal commuter, a man wearing a crossbody bag, calmly walking through a doorway cut into a frozen ocean wave, with impossible physics and cinematic light."; (3) seed 0: "A red panda curled up asleep on a mossy branch, soft morning fog in a bamboo forest."; (4) seed 1: "A lone wooden cabin on the shore of a frozen lake under a sky full of green and violet northern lights, reflections on the ice, long exposure.".

Figure 22, rows top to bottom. (1) seed 46: "A triangular purple flower pot. A purple flower pot in the shape of a triangle."; (2) seed 43: the same prompt.

Figure 26, rows top to bottom. (1) seed 42: "cat astronaut, in space, psychedelic, cosmic"; (2) seed 0: "Cinque Terre-style Italian coastal village with colorful cliff houses, turquoise sea, boats, laundry lines and warm afternoon light."; (3) seed 0: "A towering waterfall plunging into a jungle gorge, a rainbow in the spray, tiny rope bridge crossing in the foreground, lush ferns."; (4) seed 0: "A floating island with a waterfall pouring off its edge into the clouds below, a small castle on top, airships in the sky, epic matte painting.".

Figure 27, columns left to right. (1) seed 0: "High-speed macro of a water-droplet crown frozen mid-splash on a colorful surface, elegant and detailed."; (2) seed 1: "A fox reading a book under an oak tree in autumn, children's book watercolor illustration, soft washes and visible paper texture."; (3) seed 42: "a chubby goblin juggling rubber balls"; (4) seed 1: "Miniature explorer camp on a mossy log imagined as a vast landscape, with dramatic macro lighting and a tiny explorer in the scene."; (5) seed 0: "A ballerina mid-leap on an empty theater stage lit by a single spotlight, motion-blurred tulle skirt, dust particles floating in the light beam."; (6) seed 43: "close up photo of a rabbit, forest, haze, halation, bloom, dramatic atmosphere, centred, rule of thirds, 200mm 1.4f macro shot".



Seed 0: Surreal commuter, a man wearing a crossbody bag, calmly walking through a doorway cut into a frozen ocean wave, with impossible physics and cinematic light.



Seed 0: Water lily opening at dawn, frog on lily pad, dew, glowing pink petals, still pond reflection and soft mist.



Seed 1: Ornate sapphire tiara in a dim museum vitrine, blue stones and diamonds sparkling under spotlight with glass reflections.



Seed 0: Macro miniature scene: tiny fisherman on a pebble beside a blue-resin “lake” spilling from a bottle; whimsical and detailed.

Figure 21: **Paired samples at 512 pixels.** SD3.5-M, released Flow-GRPO, released DiffusionNFT and MEND at update 100, with shared initial noise; prompts from Zhou et al. (2026, Figures 11 to 14).

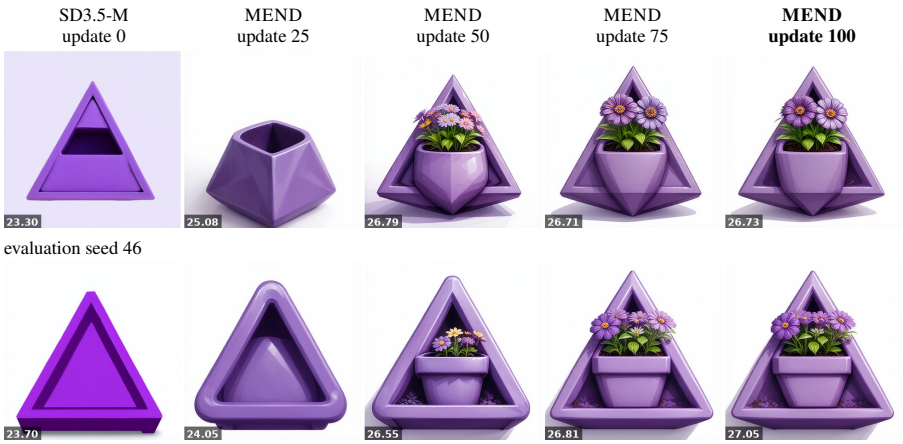

Figure 22: **One prompt along training.** "A triangular purple flower pot. A purple flower pot in the shape of a triangle." at updates 0 to 100 with fixed initial noise along each row; tiles show PickScore.

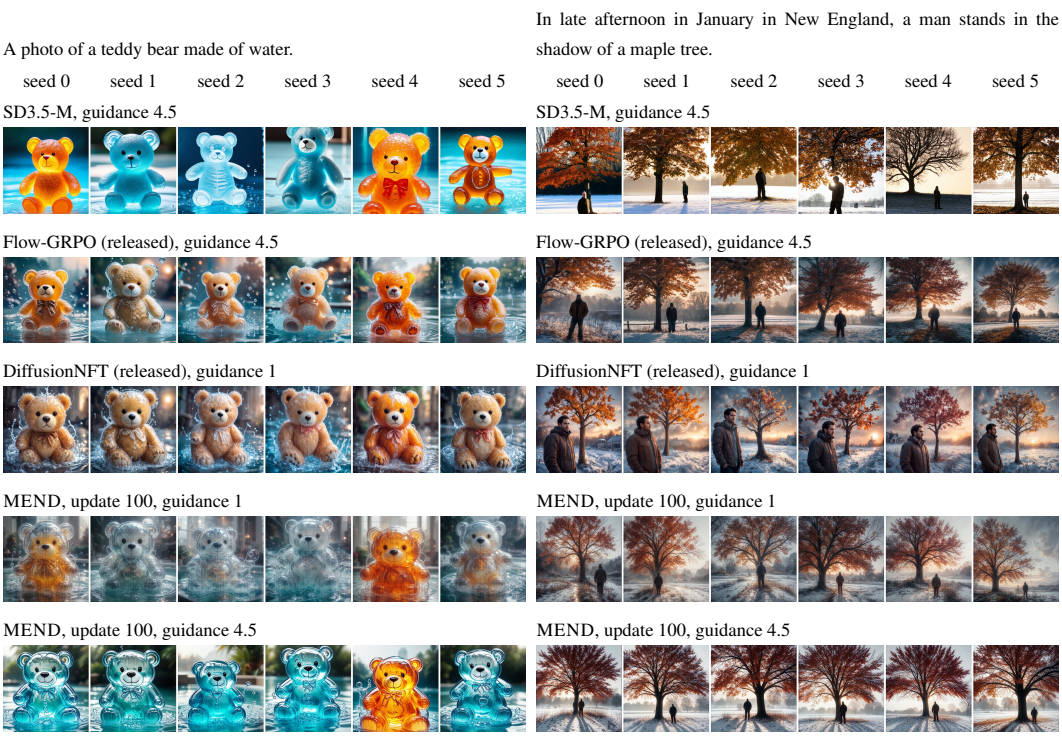

Figure 23: **Six seeds of two prompts.** Columns share initial noise across methods. Compositions stay similar while prompt fidelity mostly improves.

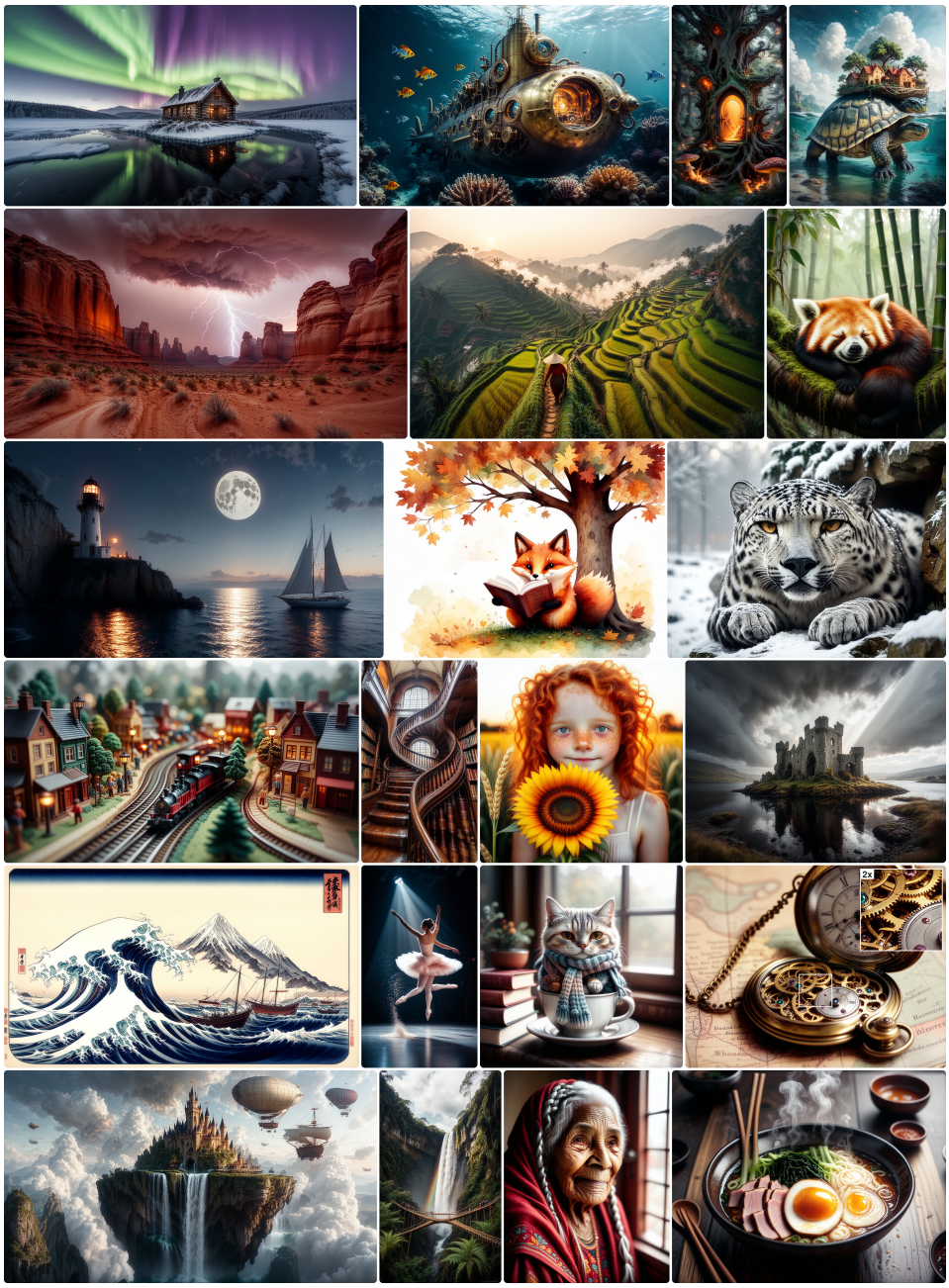

Figure 24: **MEND at 1024 pixels.** Curated tiles from the gallery prompts at native aspect ratios, without upscaling; the pocket-watch tile carries a 2× crop inset.

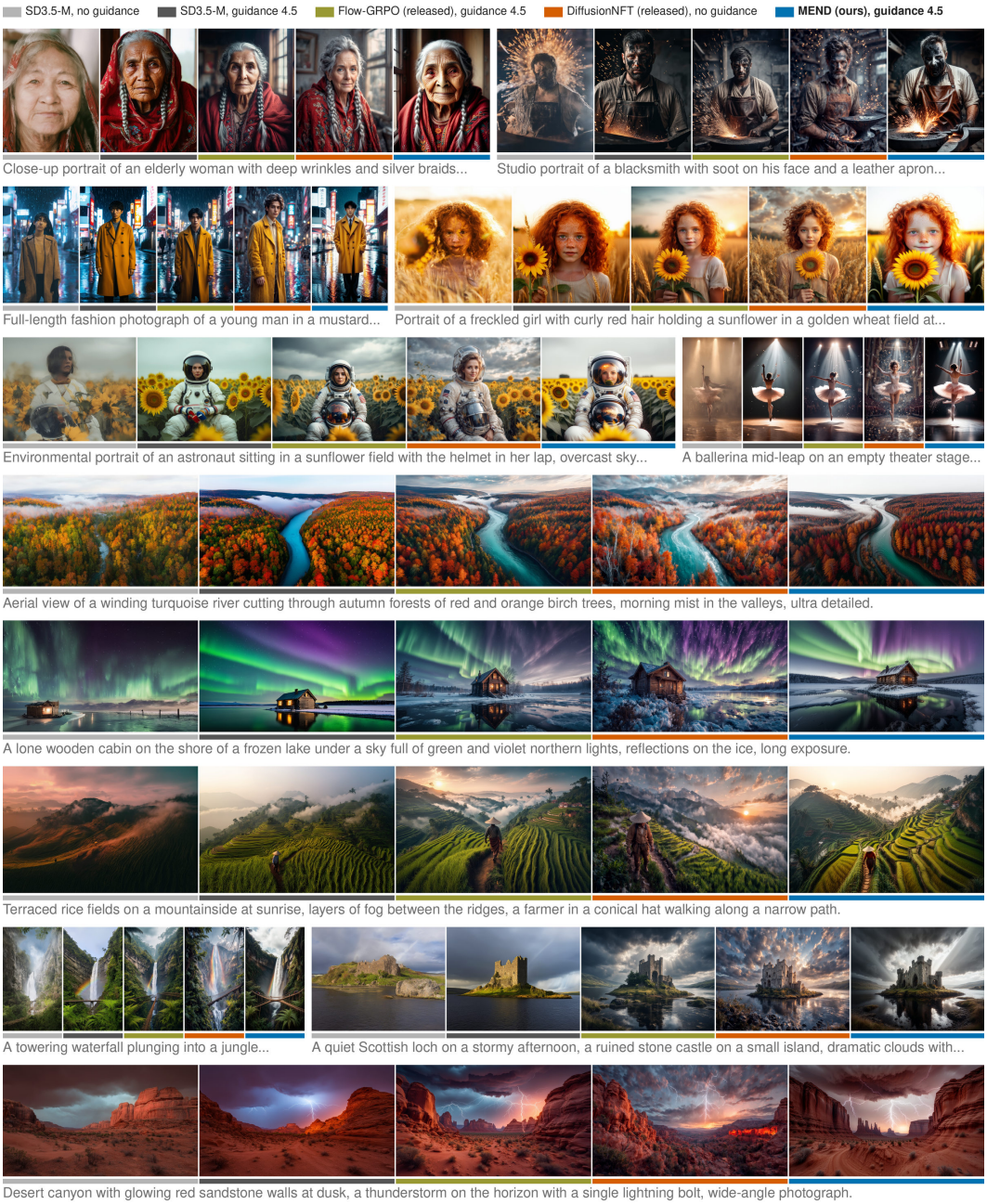

Figure 25: **Uncurated gallery sheet.** Seed 0 and shared initial noise for the base at two settings, released Flow-GRPO, released DiffusionNFT and MEND at update 100, with no quality selection.

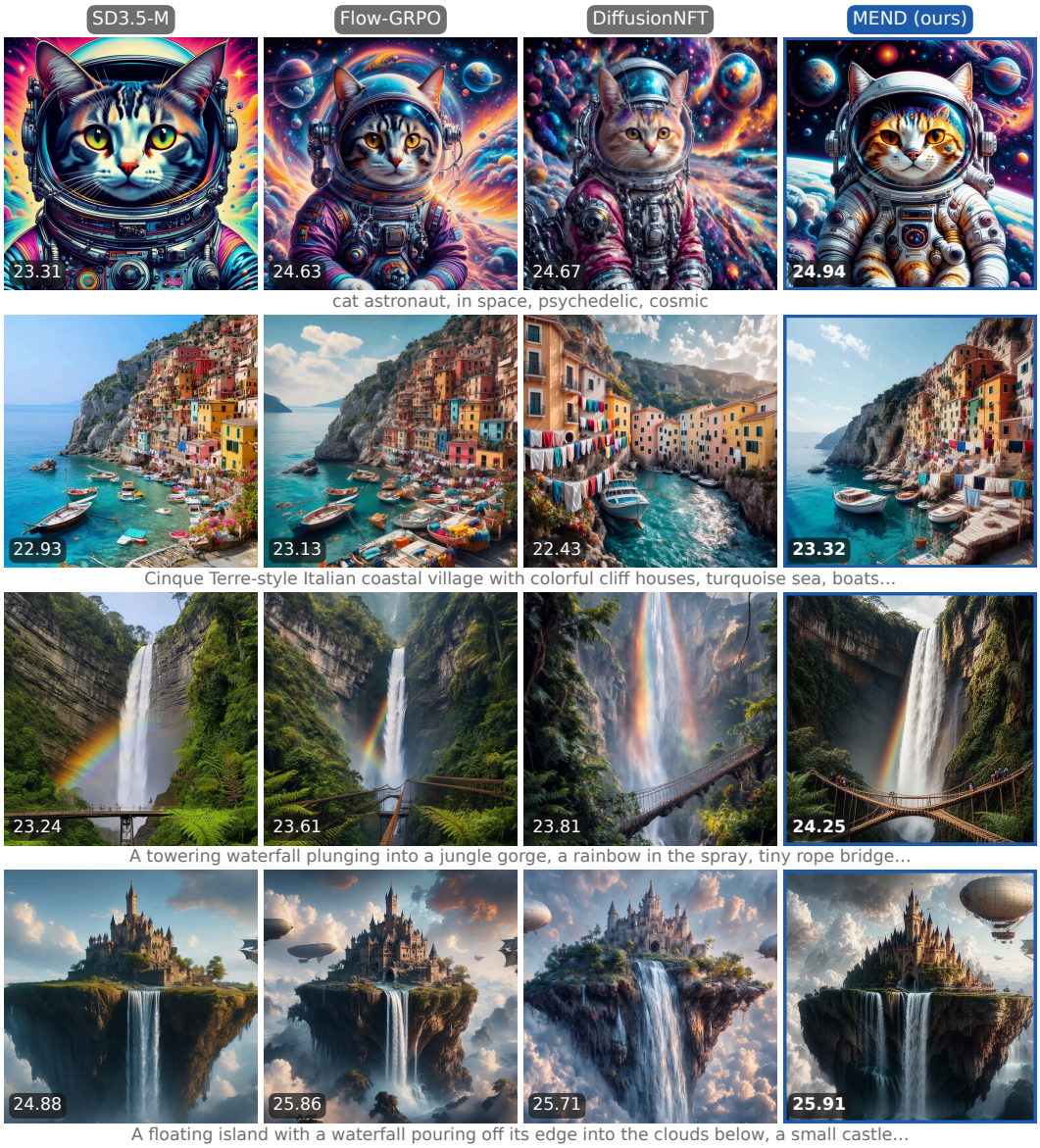

Figure 26: **More comparison rows.** Same layout as Figure 3, with PickScore on each tile.

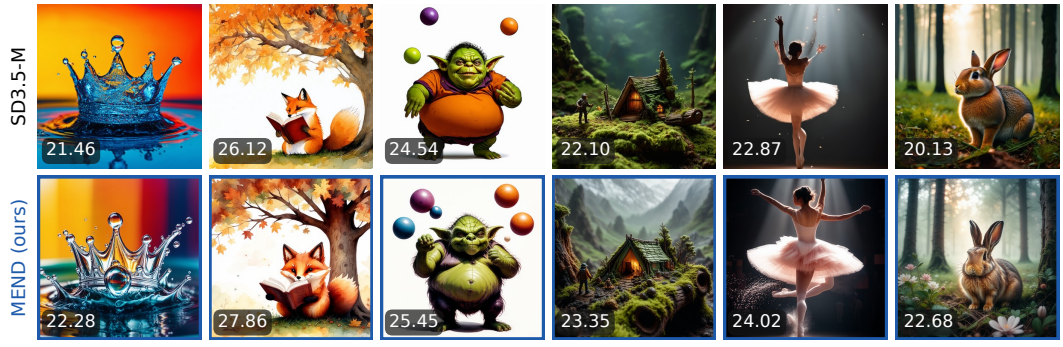

Figure 27: **More base-to-MEND pairs.** Matched prompts and initial noise, extending Figure 10.

H EXTENDED RELATED WORK

Flow-GRPO, DanceGRPO, DiffusionNFT and AWM (Liu et al., 2025; Xue et al., 2025; Zheng et al., 2026; Xue et al., 2026) reweight the model's own samples and, with a KL penalty of weight β, target the tilt $\pi_\beta \propto \pi\, e^{R/\beta}$; ReFL, DRaFT and AlignProp (Xu et al., 2023; Clark et al., 2024; Prabhudesai et al., 2023) move every sample along ∇R by an amount set by the learning rate. Few of these papers report diversity: FDFO (McAllister et al., 2026), ReNFT (Bao et al., 2026) and Adjoint Matching (Domingo-Enrich et al., 2025) use DreamSim, and ORW-CFM-W2 (Fan et al., 2025) and DMDR (Jiang et al., 2026a) use CLIP-embedding and LPIPS distances. Selection methods keep the best of N samples (Dong et al., 2023), fit samples found by a test-time search (Lee et al., 2026), accept or reject at an intermediate noise level (Anil et al., 2026) or rank fresh candidates (Jiang et al., 2026b); inference-time guidance (Bansal et al., 2024; Chung et al., 2023; He et al., 2024; Meng et al., 2022) applies the reward gradient to the sampler instead of to training targets.