DOHF: ONLINE DIFFUSION FINE-TUNING WITH DOOB'S h-TRANSFORM GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward-based diffusion fine-tuning faces practical challenges when desirable outcomes are rare or conditioning corrections are costly to estimate. In this work, we propose **Diffusion Online h-guidance Fine-tuning (DOHF)**, which turns Doob's h-transform into a practical online training algorithm. **DOHF** assigns optimality weights to generated samples, estimates the normalized local correction $\nabla \log h$ under the current rollout policy, and distills it directly into the generative model. Theoretically, we characterize the population-optimal DiffusionNFT update as well as the various classfier free guidance methods through a unified h-transform perspective. Methodologically, our framework accommodates black-box and non-differentiable rewards without additional network evaluations. We further show improved alignments under three empirical scenarios. Our work demonstrates how adapting probabilistic conditioning through inexpensive estimation and iterative distillation can improve generative learning across statistical sampling and visual generation.

1 INTRODUCTION

Doob's h-transform provides a classical probabilistic mechanism for conditioning a Markov process on a terminal event or reweighting its terminal distribution (Doob, 1957). Given a terminal weight $w(X_0)$, the associated function $h(x_t, t) = \mathbb{E}[w(X_0) \mid X_t = x_t]$ translates terminal information into a local correction of the intermediate dynamics through $\nabla \log h$. This perspective has recently become increasingly relevant to diffusion models. DEFT (Denker et al., 2024) formulates conditional generation through a generalized h-transform and learns an auxiliary network. Related work studies diffusion guidance under hard constraints (Guo et al., 2026b). More recently, DOIT (Zhu et al., 2026) uses Doob's h-transform to construct a training-free adaptation procedure. Together, these developments suggest a broader opportunity: using probabilistic conditioning to design practical improvements to generative algorithms across statistical sampling and large-scale visual generation. Empirically, however, sparse signals may provide limited information (Denker et al., 2026); exact continuation estimates are expensive (Zhu et al., 2026); and repeated reward-based updates can concentrate the generator excessively (Fan et al., 2025).

For a current rollout policy p^{old} and a general reward weight w, the desired terminal distribution is

$$p^{\mathrm{ft}}(x_0) \propto w(x_0) p^{old}(x_0),$$

while its intermediate score differs from that of the current model by exactly $\nabla_{x_t} \log h(x_t, t)$.

Existing h-transform approaches use this correction for inference-time guidance (Kim et al., 2025; Zhu et al., 2026), typically relative to a fixed reference model. This can become restrictive when the high-reward region is rare under the original proposal: the trajectories used to estimate the correction may provide little information precisely where adaptation is most needed. This motivates an online procedure in which the h-function itself is defined and estimated under the evolving rollout policy, and the resulting local correction is progressively distilled into the generative model.

In this work, we propose **Diffusion Online h-guidance Fine-tuning (DOHF)**. At each iteration, samples from the current rollout policy are assigned an optimality weight $w(X_0)$, from which we estimate the corresponding $\nabla \log h$ and distill the resulting correction directly into the generative model. This online procedure progressively shifts the rollout distribution toward more in-

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.

formative regions, while our simulation-based estimator remains applicable to black-box and non-differentiable rewards. The same formulation naturally accommodates continuous rewards, hard or soft constraints, and relative optimality objectives.

We further establish a precise connection to DiffusionNFT (Zheng et al., 2026) and identify its mixture coefficient, in which h exposes the state-dependent scaling between the population update and the conditioned velocity. This derivation gives an explicit interpretation of the normalization used by **DOHF**. We have also experimented **DOHF** on an imbalanced Gaussian mixture, contingency tables and SD3.5-Medium. All the experiments demonstrate the empirical significances of our framework: local corrections can drive rare-mode adaptation, support soft marginal matching in structured outputs, and improve visual generation with gains on an initial compositional evaluation.

Our main contributions are:

- **Unifying theoretical interpretation across settings.** We provide a population-level characterization of DiffusionNFT, showing that its update is an attenuated version of the exact h-transform, recovered only under the state-dependent choice $\beta = 2h(x_t, t, c)$. Since **DOHF** estimates h explicitly, it realizes this state-dependent strength, while allowing for an inhomogeneous β. Additionally, we have verified it that our formulation is compatible with classifier-free guidance, a design not available to forward-matching baselines.
- **Transferring probabilistic methodology into online fine-tuning.** We turn the terminal weight into a local velocity correction and regress it directly into the generative model, re-estimating h at every iteration. Our simulation-based estimator reuses the velocity at the current step without additional network evaluations, and w need not be differentiable. Estimating h and improving the rollout distribution are two sides of one update: each iteration re-collects trajectories under the tilted policy, so the correction is estimated where adaptation is most needed.
- **Demonstrating empirical significance with three experiments.** We achieve rare-mode concentration in the Gaussian experiment with fewer update iterations than the tested frozen-guidance baseline. For contingency tables, smooth weights and a frozen proposal enable soft marginal adaptation after the EMA variant exhibited collapse. On SD3.5-Medium, fine-tuning improves reward alignment and supports further CFG gains on GenEval. Together, these results demonstrate how adapting the conditioning construction can strengthen generative algorithms across statistical and visual tasks.

Related Work. Existing h-transform methods provide foundations for our work. DEFT (Denker et al., 2024), hard-constraint guidance (Guo et al., 2026b), and Doob's Matching (Chang et al., 2026) learn conditioning corrections, while DOIT (Zhu et al., 2026) estimates them through simulation. SVDD (Li et al., 2024), twisted SMC (Wu et al., 2024), and Feynman-Kac steering (Singhal et al., 2025) offer related inference-time approaches. Tilt Matching (Potaptchik et al., 2025), Iterative Tilting (Pachebat et al., 2025), and Iterative Importance Fine-tuning (Denker et al., 2026) connects tilted distributions with iterative learning.

In parallel, a rapidly growing literature studies reward-based fine-tuning of diffusion models (Uehara et al., 2024; Winata et al., 2025). Policy-based approaches formulate the reverse generation process as a policy and optimize it through RL, including DDPO (Black et al., 2024), FlowGRPO (Liu et al., 2025), DanceGRPO (Xue et al., 2025), DPOK (Fan et al., 2023), and MixGRPO (Li et al., 2026). Sampler stochasticity and the resulting training-inference gap are studied in (Sheng et al., 2025). ReFL (Xu et al., 2023) and DRaFT (Clark et al., 2024) directly differentiate reward signals, while DiffusionDPO (Wallace et al., 2023) learns from preference pairs. Adjoint Matching (Domingo-Enrich et al., 2025; Guo et al., 2026a) and Score as Action (Zhao et al., 2024; 2025) cast fine-tuning as stochastic optimal control. Gradient-based construction (Zhou et al., 2026) and flow-map alignment (Holderrieth et al., 2026; Potaptchik et al., 2026) provide alternatives when reward derivatives are available. Regression-based approaches include Advantage Weighted Matching (Xue et al., 2026) and FlowAWR (Fu et al., 2026), while online reward-weighted flow matching (Fan et al., 2025) directly reweights newly generated samples. Reward Score Matching (Lee et al., 2026), a path-space perspective (Xu et al., 2026), and reward-based velocity matching (Choi et al., 2026) clarify the shared structure of these updates.

2 PRELIMINARIES

Throughout, let c denote an optional conditioning context (e.g. a text prompt, a market covariate, or $c = \emptyset$ if unconditional). We suppress c from the notation below whenever it plays no special role. A pre-trained diffusion/flow model generates

$$X_0 \sim p^{\text{base}}(x_0 \mid c).$$

Moreover, we generate samples online, so we define the policy for rolling out is $p^{old}(x_0 \mid c)$.

2.1 A UNIFIED OPTIMALITY WEIGHT

We define optimality through a single weight function

$$w : \mathcal{X} \to [0, 1], \qquad \mathbb{E}_{p^{old}}[w(X_0)] > 0.$$

We introduce an optimality variable (Levine, 2018; Zheng et al., 2026) defining whether the generated X_0 is optimal or not:

$$O := \mathbf{1}_{\{U \le w(X_0)\}},$$

where $U \sim \text{Unif}(0, 1)$ is an auxiliary uniform random variable independent of X_0. Intuitively, X_0 with higher weight $w(X_0)$ is more likely to be optimal. Then $\mathbb{P}(O = 1 \mid X_0 = x_0) = w(x_0)$, and conditioning the base model on $\{O = 1\}$ gives the fine-tuned target

$$p^{\text{ft}}(x_0 \mid c) := p^{old}(x_0 \mid O = 1, c) = \frac{w(x_0)}{\mathbb{P}(O = 1 \mid c)}\, p^{old}(x_0 \mid c) \;\propto\; w(x_0)\, p^{old}(x_0 \mid c).$$

Every choice of $w \in [0, 1]$ therefore corresponds to a genuine conditioning event. Note also that w and Cw ($C > 0$) induce the same p^{ft}, so any normalization on w has no effect.

Examples. (i) *Exponential tilting* (Domingo-Enrich et al., 2025): $w(x_0) = \exp\big(\frac{r(x_0,c) - r_{\max}(c)}{\tau}\big)$, temperature $\tau > 0$. (ii) *Hard threshold* (Guo et al., 2026b): $w(x_0) = \mathbf{1}_{\{r(x_0,c) \ge z\}}$, giving $p^{\text{ft}} \propto p^{old}\, \mathbf{1}_{\{r \ge z\}}$. (iii) *Sigmoid relaxation*: $w(x_0) = \sigma\big(\frac{r(x_0,c) - z}{\beta}\big)$, $\sigma(u) = (1 + e^{-u})^{-1}$. Note that it recovers the hard threshold when $\beta \to 0^+$. (iv) *Loss/statistic-driven weights* (Kim et al., 2025): $w(x_0) = \psi(\ell(x_0))$ for a nondecreasing ψ, e.g. extreme summary statistics of x_0 or loss of data x_0 defined by upstream models.

2.2 THE h-FUNCTION

Define h-function as the probability of optimality given tuple (x_t, t, c) as

$$h(x_t, t, c) := \mathbb{P}(O = 1 \mid X_t = x_t, c) = \mathbb{E}\big[w(X_0) \mid X_t = x_t, c\big].$$

In particular $h(x_0, 0, c) = w(x_0)$. By Bayes' Rule, the fine-tuned marginal at time t is

$$p_t^{\text{ft}}(x_t \mid c) = p_t^{old}(x_t \mid O = 1, c) = \frac{1}{Z(c)}\, p_t^{old}(x_t \mid c)\, h(x_t, t, c), \tag{2.1}$$

where $Z(c)$ does not depend on x_t. Hence

$$\nabla_{x_t} \log p_t^{\text{ft}}(x_t \mid c) = \nabla_{x_t} \log p_t^{old}(x_t \mid c) + \nabla_{x_t} \log h(x_t, t, c), \tag{2.2}$$

so the score correction for fine-tuning is $\nabla_{x_t} \log h(x_t, t, c) = \nabla_{x_t} \log \mathbb{E}\big[w(X_0) \mid X_t = x_t, c\big]$.

2.3 FLOW MATCHING

Consider Gaussian flow matching

$$X_t = (1 - t)X_0 + tX_1, \qquad X_1 \sim \mathcal{N}(0, I), \qquad t \in [0, 1],$$

Let $v^{old}(x, t, c)$ denote the velocity field corresponding to old policy. For this Gaussian interpolation, the velocity and score are related (see Appendix A for derivation) through

$$v(x, t, c) = -\frac{x + t\nabla_x \log p_t(x \mid c)}{1 - t}.$$

By Equation 2.2, the corresponding fine-tuned velocity provides a learning target:

$$v^{\text{ft}}(x, t, c) = v^{old}(x, t, c) - \frac{t}{1 - t}\nabla_x \log h(x, t, c). \tag{2.3}$$

3 DIFFUSION ONLINE h-GUIDANCE FINE-TUNING

For Flow Matching, the backward process can be reformulated as a SDE (Huang et al., 2021):

$$dX_t = \left[(1+\lambda_t^2)v^{old}(X_t,t,c) + \frac{\lambda_t^2}{1-t}X_t\right]dt + \lambda_t\sqrt{\frac{2t}{1-t}}\,d\overline{B}_t, \tag{3.1}$$

where λ_t is a parameter that we can choose, and when $\lambda_t = 0$ it reverts to original ODE form. Consider discrete reverse times $0 = t_0 < t_1 < \cdots < t_L = 1$, and let $\phi_\ell^{old}(x_{t_{\ell-1}} \mid x_{t_\ell}, c)$ denote the backward Gaussian transition kernel, with mean $\mu_\ell(x_{t_\ell})$ and variance $\sigma_\ell^2 I$.

In order to do the regression 2.3, we need $\nabla_x \log h(x,t,c)$. Fortunately, we could use Monte Carlo Method to estimate it during the online fine-tuning process.

3.1 SIMULATION-BASED ESTIMATOR

In this section, we present a plug-in estimator of $\nabla_x \log h(x,t,c)$ (Zhu et al., 2026). Given $X_{t_\ell} = x_{t_\ell}$, simulate M backward trajectories $\{x_{t_{\ell-1}}^{(m)}, \ldots, x_0^{(m)}\}_{m=1}^M$ using ϕ^{old}. Define

$$W_m := w\big(x_0^{(m)}\big), \qquad G_m := \nabla_{x_{t_\ell}} \log \phi_\ell^{old}\big(x_{t_{\ell-1}}^{(m)} \mid x_{t_\ell}, c\big) = \frac{1}{\sigma_\ell^2}\big[\nabla_{x_{t_\ell}}\mu_\ell(x_{t_\ell})\big]^\top\big(x_{t_{\ell-1}}^{(m)} - \mu_\ell(x_{t_\ell})\big).$$

The plug-in estimator is

$$\widehat{\nabla \log h}_{\mathrm{A}} = \frac{\sum_{m=1}^M W_m G_m}{\big(\sum_{m=1}^M W_m\big) \vee \tau_\ell}, \tag{3.2}$$

with small truncation level $\tau_\ell > 0$ for numerical stability. To alleviate the cost by M rollouts of length ℓ, we use one-step-lookahead surrogate

$$\hat{x}_0^{(m)} = x_{t_{\ell-1}}^{(m)} - t_{\ell-1}\, v^{old}(x_{t_\ell}, t_\ell, c), \qquad x_{t_{\ell-1}}^{(m)} \sim \mathcal{N}\big(\mu_\ell(x_{t_\ell}), \sigma_\ell^2 I\big),$$

in place of $x_0^{(m)}$, at zero extra NFE (reusing $v^{old}(x_{t_\ell}, t_\ell, c)$ already computed). Moreover, it does not require w to be differentiable, which is more general for various terminal rewards mentioned in Section 2.1. If w is differentiable, we also provide an alternative estimator using finite difference method, which is provided in Appendix B.

3.2 TRAINING AND SAMPLING

We instantiate the training A for Text-to-Image fine-tuning, and for generalized task, prompt could be omitted. In terms of sample rolling-out, we use a slow-mix strategy: starts from on-policy and smooth it with exponential moving average. Note that we index time so that t=1 is pure noise and generation runs from $t_L = 1$ to $t_0 = 0$.

For sampling process, especially for Text-to-Image task, we could also implement the similar way used in Classifier-free Guidance (Ho & Salimans, 2022), as we set some proportion of the prompt to be empty in training process, in order to improve its unconditional generation. For other tasks, just use fine-tuned v_θ to sample. For generalization purpose, we provide Algorithm B used in T2I task.

3.3 POPULATION-LEVEL RELATION TO DIFFUSIONNFT

DiffusionNFT (Zheng et al., 2026) is the closest online fine-tuning method to **DOHF**. Although its positive–negative construction does not explicitly estimate an h-transform, its population optimum admits an exact characterization in terms of the same h-function.

Let

$$h(x_t,t,c) = \mathbb{E}_{p^{\mathrm{old}}}\left[w(X_0) \mid X_t = x_t, c\right].$$

DiffusionNFT parameterizes its positive and negative branches as

$$v_\theta^+ = (1-\beta)v^{\mathrm{old}} + \beta v_\theta, \qquad v_\theta^- = (1+\beta)v^{\mathrm{old}} - \beta v_\theta,$$

where $\beta > 0$ is a fixed guidance coefficient.

Algorithm A **D**iffusion **O**nline **h**-guidance **F**ine-tuning (**DOHF**)

Require: Pretrained v^{base}, prompt set $\mathcal{C}$, reward model r, schedule $\{t_i\}$, slow-mix schedule η_k

Initialize model weight $v_\theta \leftarrow v^{\text{base}}$, and sample-generation policy $v_\phi \leftarrow v_\theta$
for $k = 1, 2, \ldots$ **do**
  for prompt c in mini-batch **do**
    Sample t_ℓ uniformly; roll out $v_\phi(\cdot, \cdot, c)$ to obtain x_{t_ℓ}
    Compute $\widehat{\nabla \log h}(x_{t_\ell}, t_\ell, c)$ ▷ via 3.2 or B.1
    $v^{target} \leftarrow \mathbf{stopgradient}\Big(v_\phi(x_{t_\ell}, t_\ell, c) - \frac{t_\ell}{1-t_\ell}\,\widehat{\nabla \log h}\Big)$
    $\mathcal{L} \leftarrow \mathcal{L} + \|v_\theta(x_{t_\ell}, t_\ell, c) - v^{target}\|_2^2$
  end for
  Update θ by one step on $\nabla_\theta \mathcal{L}$
  $v_\phi \leftarrow \eta_k v_\phi + (1 - \eta_k) v_\theta$ ▷ slow-mix sample-generation policy
end for
return v_θ

Algorithm B Sampling with Classifier-Free Guidance

Require: Trained v_θ, prompt c, guidance scale $\gamma \geq 0$, schedule $\{t_i\}_{i=0}^{N}$

$x \leftarrow z \sim \mathcal{N}(0, I)$
for $i = N, N-1, \ldots, 1$ **do**
  $v_{\text{cond}} \leftarrow v_\theta(x, t_i, c), \quad v_{\text{uncond}} \leftarrow v_\theta(x, t_i, \varnothing)$
  $v \leftarrow v_{\text{uncond}} + \gamma\,(v_{\text{cond}} - v_{\text{uncond}})$
  $x \leftarrow x + v\,(t_{i+1} - t_i)$
end for
return x

Proposition. At the population optimum of the double-branch DiffusionNFT objective, the learned velocity satisfies

$$v_\theta^*(x_t, t, c) = v^{\text{old}}(x_t, t, c) - \frac{2h(x_t, t, c)}{\beta} \frac{t}{1-t} \nabla_{x_t} \log h(x_t, t, c).$$

Consequently, its population optimum coincides with the exact h-transformed velocity

$$v^{\text{ft}} = v^{\text{old}} - \frac{t}{1-t} \nabla_{x_t} \log h$$

if and only if

$$\beta = 2h(x_t, t, c).$$

A complete derivation is provided in Appendix C.

This result reveals a key distinction between the two methods. With a fixed β, DiffusionNFT effectively follows the unnormalized direction $\nabla h = h \nabla \log h$; hence, its update is attenuated in low-h regions where optimal samples are rare. **DOHF** instead directly estimates the normalized correction $\nabla \log h$ under the evolving rollout policy and distills it into the model. It therefore realizes the exact h-transformed update without requiring a state-dependent guidance hyperparameter.

4 Experiment

4.1 Gaussian Mixture

To isolate the effect of online iteration from confounders, we conduct a controlled study on a 2D synthetic problem. The base distribution is an imbalanced two-mode Gaussian mixture

$$p_{\text{data}}(x) \;=\; 0.01 \cdot \mathcal{N}\big(x;\, \mu_0,\, \Sigma\big) \;+\; 0.99 \cdot \mathcal{N}\big(x;\, \mu_1,\, \Sigma\big),$$

with $\mu_0 = (-1, -1)^\top$, $\mu_1 = (1, 1)^\top$, and $\Sigma = 0.16\,\mathbf{I}$. A rectified-flow velocity network (a small MLP) is first pretrained on samples from p_{data}.

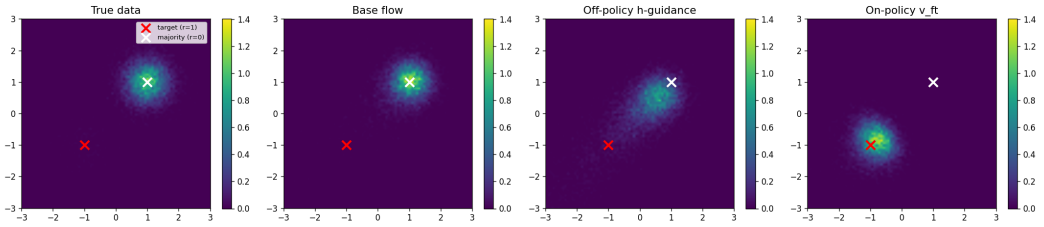

Figure 1: Visualization of on/off-policy h-transform guidance.

Method	Iterations	Minority-mode fraction	Mean $\pm$ std
True data	–	0.96%	$(0.98, 0.98) \pm (0.44, 0.44)$
Base pretrained	–	1.12%	$(0.99, 0.98) \pm (0.42, 0.41)$
Off-policy h-guidance	2000	14.88%	$(0.29, 0.25) \pm (0.73, 0.71)$
DOHF(ours)	200	**100.00%**	$(-0.82, -0.88) \pm (0.38, 0.38)$

Table 1: Fraction of samples falling in the minority cluster after fine-tuning.

The reward is the *soft posterior probability of belonging to the minority mode*:

$$r(x) \;=\; \mathbb{P}(\text{cluster} = 0 \,|\, x) \;=\; \frac{0.01 \cdot \mathcal{N}(x;\, \mu_0,\, \Sigma)}{0.01 \cdot \mathcal{N}\big(x;\, \mu_0,\, \Sigma\big) \;+\; 0.99 \cdot \mathcal{N}\big(x;\, \mu_1,\, \Sigma\big)} \;\in\; (0, 1).$$

This turns the alignment task into an extreme mode-seeking problem: the fine-tuned model should transport probability mass from the majority mode μ_1 into the minority mode μ_0.

We compare two reward-tilting strategies, both starting from the same pretrained base:

- **Off-policy h-guidance (Guo et al., 2026b).** A separate value network $h_\phi(x_t, t) \approx \mathbb{E}[r(X_0) \,|\, X_t{=}x_t]$ is regressed on rollouts drawn *exclusively* from the frozen base, then applied as guidance at inference via $\hat{v} = v_{\text{base}} - \frac{t}{1-t} \nabla \log h_\phi$.
- **DOHF(ours).** The same h-transform correction is instead distilled into the velocity network itself via online data collection.

Table 1 summarizes the outcome. Off-policy guidance takes 2000 update steps and still lifts the minority-mode fraction to some extent ($1.12\% \to 14.88\%$): most of its training rollouts still land in the majority mode. In contrast, online h-distillation reaches 100% minority-mode concentration after just 200 steps — one order of magnitude fewer updates for a much better outcome: better preserves the target distribution with closer mean $(-0.82, -0.88)$ and standard deviation $(0.38, 0.38)$. Because each iteration re-collects trajectories under tilted policy, the reward-relevant regions of state space are explored progressively as the model itself improves. For visualization, see Figure 1.

4.2 Contingency Table

We consider contingency-table generation as an application of endogenous conditional sampling under the framework of (Guo et al., 2026b). Classical references include (Diaconis & Efron, 1985; Diaconis & Sturmfels, 1998), with connections to Burnside processes and double cosets discussed in (Diaconis & Howes, 2025, Section 3). Such tables also arise in financial-network and systemic-risk applications (Albanesi & Vamossy, 2019; Greenwood et al., 2022).

Objective. Let $X = (x_{ij}) \in \mathbb{Z}_+^{m \times n}$ be contingency table with prescribed row and column sums

$$\sum_j x_{ij} = r_i, \qquad \sum_i x_{ij} = c_j. \tag{4.1}$$

In practice, these marginals may be noisy or partially observed, so we treat them as soft structural targets.

Method	Updates	Avg. Margin Error ↓	Max Margin Error ↓	W_2 vs. DG sampler ↓
Uniform	–	0.802	1.975	13.48
Off-policy	800	0.379	1.005	11.00
DOHF	**300**	**0.210**	**0.531**	**9.84**

Table 2: Comparison of margin matching and distributional similarity. The margin errors are standardized by the corresponding generated marginal standard deviations. The W_2 metrics are evaluated according to (Diaconis & Gangolli, 1995)

Given a pretrained flow model v^{base}, define

$$\mathcal{S}_2 = \sum_i \left(\sum_j x_{ij} - r_i \right)^2 + \sum_j \left(\sum_i x_{ij} - c_j \right)^2 .$$

Exact feasibility corresponds to $\mathcal{S}_2 = 0$. Since exact matches are extremely rare for large tables, directly using the indicator reward leads to high-variance h-guidance estimates, usually there is no such table in a batch satisfying $\mathcal{S}_2 = 0$. We therefore use the smooth reward

$$w(x) = \exp(-\beta \mathcal{S}_2),$$

where larger β more closely approximates the hard constraint.

Birthday–deathday contingency table. We evaluate on the classical 12×12 birthday–deathday table of 82 descendants of Queen Victoria (Diaconis & Sturmfels, 1998). The target margins are

$$r = (6, 5, 5, 12, 12, 3, 10, 7, 3, 7, 9, 3), \qquad c = (13, 4, 7, 10, 8, 4, 5, 3, 4, 9, 7, 8),$$

with total count 82, the original table is provided as Table 4.

We pre-train a flow model that generates nonnegative integer-valued 12×12 tables with total count 82; implementation details are given in Appendix D.2. Starting from the same pretrained model, we compare frozen off-policy h-guidance with **DOHF** using the reward $\exp(-\beta \mathcal{S}_2)$, where we take $\beta = 0.02$.

Among 500,000 uniformly sampled pre-training tables, none exactly satisfies all target margins, illustrating the sparsity of the target event and the necessity of using soft targets. As shown in Table 2, uniform sampling (as how we generate pre-training data) yields an average standardized margin error of 0.802 and a maximum error of 1.975. After only 300 updates, **DOHF** reduces these errors to 0.210 and 0.531, respectively, outperforming the off-policy baseline, which reaches 0.379 and 1.005 after 800 updates.

The Diaconis–Gangolli (DG) sampler (Diaconis & Gangolli, 1995) is a Markov Chain based method, generating approximately uniform contingency tables with fixed margins through repeated local 2×2 integer-preserving moves, which requires initial burn-in (in our experiment 200,000 steps are used). Although we are generating tables with perturbed margins, it is still worthwhile to check the difference between generated distributions. It is shown that **DOHF** also achieves the smallest Wasserstein distance to the DG sampler baseline among the three methods.

The advantage comes from adapting the rollout distribution toward the target region, which provides increasingly informative samples for subsequent h-transform updates. For this experiment, however, we set $v_\phi = v^{\text{base}}$ at each iteration rather than using the EMA update in Algorithm A, since the latter empirically caused collapse to nearly identical tables. This modification preserves rollout diversity while retaining the benefit of online h-distillation.

4.3 Image Fine-tuning

Our flow-based Text-to-Image model fine-tuning task is based on **SD3.5-Medium** (Esser et al., 2024) at 256×256 resolution. We demonstrate the potential of our method trained with the sum of three reward models: HPSv2.1 (Wu et al., 2023), PickScore (Kirstain et al., 2023) and CLIP-Score (Radford et al., 2021).



Figure 2: Qualitative comparison across different guidance and fine-tuning methods. The prompts are taken from PICK-A-PIC (Kirstain et al., 2023).

We will evaluate our method and other baselines on both in-domain and out-of-domain metrics: IMAGEREWARD (Xu et al., 2023), AESTHETIC SCORE (Schuhmann et al., 2022), GENEVAL (Ghosh et al., 2023) and pair-wise LPIPS (Zhang et al., 2018). Note that we used average pair-wise LPIPS per prompt to measure image diversity. We aim to show that rolling out online is not only more efficient compared to off-policy fashion, but also provide a higher reward, combined with better semantic alignment. For Qualitative comparison, refer to Figure 2.

The training and testing prompt set is the ones used in Pick-a-Pic (Kirstain et al., 2023). We finetuned with LoRA($\alpha = 32, r = 32$). Each iteration consists of 28 prompts, and generates $M = 8$ images per prompt via one-step-lookahead surrogate discussed in Section 3.1. For both training and testing roll-outs, 40 time steps are applied. Additional details are provided in Appendix E.

As shown in Table 3, our **DOHF** methodology works very well even without classifier-free-guidance (CFG) (Ho & Salimans, 2022) using only 2k iterations, surpassing FlowGRPO (Liu et al., 2025) on most of the metrics using 9k iterations. Moreover, it performs slightly better than DiffusionNFT (Zheng et al., 2026) overall in 2k iterations. To better explain their relationship, detailed discussion are provided in Appendix C.

However, our online h-guidance has CFG design which is not available in DiffusionNFT. During training, we set 10% of the prompts to be empty, in order to enforce model to have better unconditional generation, thus $v_\theta(\cdot, \cdot, c) - v_\theta(\cdot, \cdot, \emptyset)$ acts like guidance. During inference, velocity field is $v_\theta(\cdot, \cdot, \emptyset) + \gamma\big(v_\theta(\cdot, \cdot, c) - v_\theta(\cdot, \cdot, \emptyset)\big)$, where $\gamma = 4$ is the default guidance scale. It is shown in Table 3 that with guidance, all the quality scores increase to some extent, at little cost of diversity. It is worth noting that GENEVAL, the out-of-domain rule-based reward measuring how accurately generated images satisfy compositional prompts, increase significantly from 44.27% to 62.52%, showing its potential in semantic alignment.

Model	#Iter	GenEval	PickScore	ClipScore	HPSv2.1	Aesthetic	ImgRwd	LPIPS
SD3.5-M w/o CFG	–	4.49%	0.179	0.178	0.150	4.513	-1.809	0.771
SD3.5-M w/ CFG	–	41.35%	0.203	0.266	0.248	5.428	0.250	0.674
SD3.5-L w/ CFG	–	25.00%	0.193	0.243	0.208	4.919	-0.452	0.733
DOHF w/o CFG	2k	44.27%	<u>0.216</u>	<u>0.264</u>	<u>0.327</u>	**6.327**	1.115	0.538
DOHF w/ CFG	2k	**62.52%**	**0.218**	**0.277**	**0.336**	<u>6.240</u>	**1.329**	0.521
Off-policy Guidance	2k	19.52%	0.184	0.201	0.177	4.739	-1.320	**0.746**
	8k	38.96%	0.197	0.240	0.249	5.422	0.190	<u>0.655</u>
DiffusionNFT	2k	<u>49.39%</u>	0.214	0.261	0.305	6.151	<u>1.118</u>	0.525
FlowGRPO	2k	35.89%	0.205	0.252	0.272	5.924	0.636	0.602
	9k	45.66%	0.211	0.259	0.302	6.162	1.020	0.588

Table 3: Evaluation Results under resolution 256×256. **Bold**: best; <u>Underline</u>: second best.

5 Conclusion

We introduced **DOHF**, an online diffusion fine-tuning framework based on Doob's h-transform. Our method turns reward-based adaptation into a local score or velocity correction that can be estimated online and directly distilled into the generative model. This perspective provides a unified treatment of continuous rewards, including hard restrictions, soft constraints, and relative optimality objectives. Our work also clarifies the connection to existing methods such as DiffusionNFT.

Across rare-mode adaptation, constrained contingency table generation, and text-to-image fine-tuning, **DOHF** consistently demonstrates the benefit of adapting the rollout distribution together with the model, particularly when reward-relevant regions are difficult to sample via base model. These results suggest that online h-transform fine-tuning offers a promising alternative to both inference-time guidance and trajectory-based reinforcement learning. More broadly, the framework provides a flexible interface between probabilistic conditioning and generative-model optimization, and may be useful for a wide range of problems involving rare events, structured constraints, black-box rewards, and distributional adaptation.

AI use statement

We used large language models (LLMs) solely to improve the clarity, grammar, and organization of the manuscript. We did not use generative AI to develop the methodology, formulate mathematical proofs, design experiments, implement methods, or interpret experimental results. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

Reproducibility statement

We describe the experimental setups, datasets, and base models for all three tasks (Gaussian-mixture adaptation, contingency-table generation, and text-to-image fine-tuning) in Section 4. The theoretical foundations are supported by relevant citations, with detailed derivations given in Appendix A. Algorithms A and B provide concrete training and sampling implementation guidance in the main text, with additional implementation details in Appendix E. In addition to the descriptions that support the reproducibility of our experiments, we will publicly release the simulation and training code after the conference review period.

References

Stefania Albanesi and Domonkos F. Vamossy. Predicting consumer default: A deep learning approach. Working Paper 26165, National Bureau of Economic Research, August 2019. URL https://www.nber.org/papers/w26165.

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

Jinyuan Chang, Chenguang Duan, Yuling Jiao, Yi Xu, and Jerry Zhijian Yang. Inference-time alignment for diffusion models via variationally stable doob's matching, 2026. URL https://arxiv.org/abs/2601.06514.

Jaemoo Choi, Wei Guo, Yuchen Zhu, Arash Vahdat, Molei Tao, Julius Berner, and Yongxin Chen. Scaling reinforcement learning for diffusion models via velocity matching, 2026. URL https://arxiv.org/abs/2608.23664.

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

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. DEFT: Efficient fine-tuning of diffusion models by learning the generalised h-transform. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Alexander Denker, Shreyas Padhy, Francisco Vargas, and Johannes Hertrich. Iterative importance fine-tuning of diffusion models, 2026. URL https://arxiv.org/abs/2502.04468.

Persi Diaconis and Bradley Efron. Testing for independence in a two-way table: New interpretations of the chi-square statistic. *The Annals of Statistics*, 13(3):845–874, 1985. doi: 10.1214/aos/1176349634. URL https://projecteuclid.org.

Persi Diaconis and Anil Gangolli. Rectangular arrays with fixed margins. In *Discrete Probability and Algorithms*, volume 72 of *IMA Volumes in Mathematics and Its Applications*, pp. 15–41. Springer, 1995.

Persi Diaconis and Michael Howes. Random sampling of contingency tables and partitions: Two practical examples of the burnside process. *Statistics and Computing*, 35(6), August 2025. ISSN 1573-1375. doi: 10.1007/s11222-025-10708-5. URL http://dx.doi.org/10.1007/s11222-025-10708-5.

Persi Diaconis and Bernd Sturmfels. Algebraic algorithms for sampling from conditional distributions. *The Annals of Statistics*, 26(1):363–397, 1998.

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, 2025. URL https://arxiv.org/abs/2409.08861.

J. L. Doob. Conditional brownian motion and the boundary limits of harmonic functions. *Bulletin de la Société Mathématique de France*, 85:431–458, 1957. doi: 10.24033/bsmf.1494.

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

Jiajun Fan, Shuaike Shen, Chaoran Cheng, Yuxin Chen, Chumeng Liang, and Ge Liu. Online reward-weighted fine-tuning of flow matching with wasserstein regularization, 2025. URL https://arxiv.org/abs/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, 2023. URL https://arxiv.org/abs/2305.16381.

Zheming Fu, Ruizhe He, Wei Shang, Xiaoxiao Ma, Lei Wang, Chang Liu, and Siming Fu. Flowawr: Online adaptive flow reinforcement via advantage-weighted rectification, 2026. URL https://arxiv.org/abs/2606.30376.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment, 2023. URL https://arxiv.org/abs/2310.11513.

Robin Greenwood, Samuel G. Hanson, Andrei Shleifer, and Jakob Ahm Sørensen. Predictable financial crises. *The Journal of Finance*, 77(2):863–921, 2022. doi: 10.1111/jofi.13105.

Zhengyi Guo, Jiayuan Sheng, David D. Yao, and Wenpin Tang. Improved techniques for fine-tuning flow models via adjoint matching: a deterministic control pipeline, 2026a. URL https://arxiv.org/abs/2605.06583.

Zhengyi Guo, Wenpin Tang, and Renyuan Xu. Conditional diffusion guidance under hard constraint: A stochastic analysis approach. *arXiv preprint arXiv:2602.05533*, 2026b. URL https://arxiv.org/abs/2602.05533.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps, 2026. URL https://arxiv.org/abs/2602.05993.

Chin-Wei Huang, Jae Hyun Lim, and Aaron Courville. A variational perspective on diffusion-based generative models and score matching. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2106.02808.

Heasung Kim, Taekyun Lee, Hyeji Kim, and Gustavo De Veciana. Generating informative samples for risk-averse fine-tuning of downstream tasks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kfB5Ciz2XZ.

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, 2023. URL https://arxiv.org/abs/2305.01569.

Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models, 2026. URL https://arxiv.org/abs/2604.17415.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018. URL https://arxiv.org/abs/1805.00909.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, and Liefeng Bo. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde, 2026. URL https://arxiv.org/abs/2507.21802.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding, 2024. URL https://arxiv.org/abs/2408.08252.

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, 2025. URL https://arxiv.org/abs/2505.05470.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, June 2025. ISSN 2731-5398. doi: 10.1007/s11633-025-1562-4. URL http://dx.doi.org/10.1007/s11633-025-1562-4.

Jean Pachebat, Giovanni Conforti, Alain Durmus, and Yazid Janati. Iterative tilting for diffusion fine-tuning, 2025. URL https://arxiv.org/abs/2512.03234.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S. Albergo. Tilt matching for scalable sampling and fine-tuning, 2025. URL https://arxiv.org/abs/2512.21829.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael S. Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment, 2026. URL https://arxiv.org/abs/2601.14430.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. URL https://arxiv.org/abs/2210.08402.

Jiayuan Sheng, Hanyang Zhao, Haoxian Chen, David D. Yao, and Wenpin Tang. Understanding sampler stochasticity in training diffusion models for RLHF. *arXiv preprint arXiv:2510.10767*, 2025.

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

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization, 2023. URL https://arxiv.org/abs/2311.12908.

Genta Indra Winata, Hanyang Zhao, Anirban Das, Wenpin Tang, David D. Yao, Shi-Xiong Zhang, and Sambit Sahu. Preference tuning with human feedback on language, speech, and vision tasks: a survey. *J. Artificial Intelligence Res.*, 82:2595–2661, 2025.

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models, 2024. URL https://arxiv.org/abs/2306.17775.

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, 2023. URL https://arxiv.org/abs/2306.09341.

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

Yixian Xu, Yuanrui Zhang, Shengjie Luo, Liwei Wang, and Di He. Designing reinforcement learning for diffusion models: A unified path-space view, 2026. URL https://arxiv.org/abs/2608.14430.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models, 2026. URL https://arxiv.org/abs/2509.25050.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Hanyang Zhao, Haoxian Chen, Ji Zhang, David D. Yao, and Wenpin Tang. Scores as Actions: a framework of fine-tuning diffusion models by continuous-time reinforcement learning. *arXiv preprint arXiv:2409.08400*, 2024.

Hanyang Zhao, Haoxian Chen, Ji Zhang, David D. Yao, and Wenpin Tang. Score as Action: fine tuning diffusion generative models by continuous-time reinforcement learning. In *ICML*, 2025.

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

Wei Zhou, Xiongwei Zhu, Lingdong Kong, Bo Chen, Lei Zhang, Yongyuan Liang, Xiaoxia Hou, Ye Tian, Xian Sun, Yingshuo Wang, Linfeng Li, Shengqiong Wu, Leigang Qu, Feng Li, Wei Liu, Julian McAuley, and Tat-Seng Chua. On-policy self-distillation in diffusion models, 2026. URL https://arxiv.org/abs/2608.24646.

Qijie Zhu, Zeqi Ye, Han Liu, Zhaoran Wang, and Minshuo Chen. Training-free adaptation of diffusion models via doob's h-transform, 2026. URL https://arxiv.org/abs/2602.16198.

A Relation between score and velocity

Consider the rectified-flow interpolation

$$X_t = (1-t)X_0 + t\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

The corresponding velocity field is

$$v(x,t) = \mathbb{E}[\epsilon - X_0 \mid X_t = x]\,.$$

Conditioned on X_0, we have

$$X_t \mid X_0 \sim \mathcal{N}\big((1-t)X_0, t^2 I\big)\,,$$

and hence

$$\nabla_x \log p_t(x \mid X_0) = -\frac{x-(1-t)X_0}{t^2} = -\frac{\epsilon}{t}.$$

Using the score marginalization identity,

$$\nabla_x \log p_t(x) = \mathbb{E}[\nabla_x \log p_t(x \mid X_0) \mid X_t = x]\,,$$

we obtain

$$\nabla_x \log p_t(x) = -\frac{1}{t}\mathbb{E}[\epsilon \mid X_t = x]\,.$$

Therefore,

$$\mathbb{E}[\epsilon \mid X_t = x] = x + (1-t)\mathbb{E}[\epsilon - X_0 \mid X_t = x] = x + (1-t)v(x,t).$$

Substituting this into the previous expression gives

$$\nabla_x \log p_t(x) = -\frac{x + (1-t)v(x,t)}{t}.$$

Therefore, we have

$$v(x,t) = -\frac{x + t\nabla_x \log p_t(x)}{1-t}.$$

B Alternative Estimator of $\nabla_x \log h(x,t,c)$:

Approximate $h(x,t,c) \approx w(\hat{x}_0(x,t,c))$ via the Tweedie point estimate

$$\hat{x}_0(x,t,c) = x - t\, v^{old}(x,t,c),$$

so that $\nabla_x \log h(x,t,c) \approx \big[\partial_x \hat{x}_0(x,t,c)\big]^\top u$, with $u := \nabla_{\hat{x}_0} \log w(\hat{x}_0(x,t,c))$. Moreover, we should avoid the Jacobian $\partial_x \hat{x}_0$ via a finite-difference step of size $\epsilon > 0$:

$$\widehat{\nabla \log h}_{\mathrm{B}} = \frac{1}{1-t}\left[u + \frac{t^2}{\epsilon}\Big(\nabla_x \log p_t^{old}(x + \epsilon u \mid c) - \nabla_x \log p_t^{old}(x \mid c)\Big)\right], \tag{B.1}$$

where $\nabla_x \log p_t^{old}(x \mid c) = -\frac{x+(1-t)v^{old}(x,t,c)}{t}$. This method (Kim et al., 2025) doesn't require M rollouts in Estimator 3.2, only 2 evaluations of v^{old} plus one backward pass through w. However, it requires w to be differentiable.

C Relation to Diffusion Negative-aware Fine-Tuning (NFT)

DiffusionNFT (Zheng et al., 2026) arrives at a guidance mechanism built from the optimality variable $r(x_0, c) = p(O{=}1 \mid x_0, c)$, which is exactly our weight $w(x_0)$. We show that its guidance direction, and both its single-branch and double-branch training objectives, are special cases of the h-transform framework developed above.

C.1 Distribution split coefficient $\alpha(x_t)$ is equvalent to $h(x_t, t, c)$

DiffusionNFT defines $\pi^+(x_0 \mid c) := \pi^{old}(x_0 \mid O{=}1, c)$ and $\pi^-(x_0 \mid c) := \pi^{old}(x_0 \mid O{=}0, c)$. It can be shown that the posterior distribution of old policy can be seperated as

$$\pi^{old}_{0|t}(x_0 \mid x_t, c) = \alpha(x_t)\pi^+_{0|t}(x_0 \mid x_t, c) + [1 - \alpha(x_t)]\pi^-_{0|t}(x_0 \mid x_t, c),$$

where

$$\alpha(x_t) := \frac{\pi^+_t(x_t \mid c)}{\pi^{old}_t(x_t \mid c)} \cdot \mathbb{E}_{\pi^{old}(x_0|c)}[w(x_0)].$$

Also,

$$v^{old}(x_t, t, c) = \alpha(x_t)v^+(x_t, t, c) + [1 - \alpha(x_t)]v^-(x_t, t, c). \tag{C.1}$$

Since $\pi^+(x_0 \mid c) = p^{old}(x_0 \mid O{=}1, c) = p^{\text{ft}}(x_0 \mid c)$ by definition, we have $\pi^+_t = p^{\text{ft}}_t$, and from equation 2.1,

$$\frac{p^{\text{ft}}_t(x_t \mid c)}{p^{old}_t(x_t \mid c)} = \frac{h(x_t, t, c)}{Z(c)}.$$

Since $\mathbb{E}_{\pi^{old}(x_0|c)}[w(x_0)] = Z(c)$, substituting gives

$$\alpha(x_t) = \frac{h(x_t, t, c)}{Z(c)} \cdot Z(c) = h(x_t, t, c).$$

As noted above, $\pi^+_t \equiv p^{\text{ft}}_t$ as distributions, hence their posteriors over x_0 given x_t coincide, and since the velocity field is a linear functional of the posterior mean,

$$v^+(x_t, t, c) = v^{\text{ft}}(x_t, t, c) = v^{\text{base}}(x_t, t, c) - \frac{t}{1-t}\nabla_x \log h(x_t, t, c).$$

C.2 The Guidance Direction Δ and v^- as Anti-Guidance

According to equation C.1, v^{old}, v^+, v^- are on the same line. DiffuionNFT defines velocity improvement direction Δ as:

$$\Delta := \alpha(x_t)\,[v^+ - v^{old}] = [1 - \alpha(x_t)]\,[v^{old} - v^-].$$

Using $\alpha = h$ and $v^+ - v^{old} = v^{\text{ft}} - v^{old} = -\frac{t}{1-t}\nabla_x \log h$,

$$\Delta = h \cdot \Big(-\frac{t}{1-t}\nabla_x \log h \Big) = -\frac{t}{1-t}\, h\, \nabla_x \log h = -\frac{t}{1-t}\nabla_x h,$$

using $h\nabla \log h = \nabla h$. That is, Δ is the (unnormalized) gradient $\nabla_x h$, rescaled by $-\frac{t}{1-t}$.

For v^-: by the same construction with $w^-(x_0) := 1 - w(x_0)$, $\pi^- = p^{old}(\cdot \mid O{=}0, c)$ has $h^-(x_t, t, c) = 1 - h(x_t, t, c)$, and

$$v^-(x_t, c, t) = v^{old}(x_t, t, c) - \frac{t}{1-t}\nabla_x \log \big(1 - h(x_t, t, c)\big).$$

Substituting into the first form of Δ, with $\nabla_x \log(1 - h) = -\nabla_x h/(1 - h)$,

$$-[1 - h]\Big[v^{old} - v^-\Big] = -[1 - h] \cdot \frac{t}{1-t} \cdot \frac{\nabla_x h}{1 - h} = -\frac{t}{1-t}\nabla_x h,$$

confirming consistency. Thus v^- is precisely the h-transform velocity for the *complementary* event $\{O = 0\}$: it is not an independent construction but an "anti-guidance" field pointing away from the optimal region.

C.3 DOUBLE-BRANCH OPTIMAL VELOCITY

DiffusionNFT trains v_θ (it acts as an anchor for improvement guidance while training, and used for inference after training) via

$$\mathcal{L}(\theta) = \mathbb{E}_{c,\pi^{old}(x_0|c),t}\Big[w(x_0)\,\|v_\theta^+ - v\|_2^2 + (1 - w(x_0))\,\|v_\theta^- - v\|_2^2\Big],$$

where

$$v_\theta^+ := (1-\beta)v^{old} + \beta v_\theta, \quad v_\theta^- := (1+\beta)v^{old} - \beta v_\theta. \tag{C.2}$$

The 'good velocity' is parametrized as a linear combination between anchor v_θ and v^{old}, and 'bad velocity' is pushing v^{old} to the opposite direction from the 'good' one.

By definition $p^{\text{ft}}(x_0 \mid c) = \frac{w(x_0)}{Z(c)} p^{old}(x_0 \mid c)$, i.e.

$$w(x_0)\, p^{old}(x_0 \mid c) = Z(c)\, p^{\text{ft}}(x_0 \mid c).$$

Multiplying both sides by the forward transition kernel $\phi(x_t \mid x_0)$, which is shared by p^{old} and p^{ft}, and applying Bayes' rule $p(x_0 \mid c)\phi(x_t \mid x_0) = p_t(x_t \mid c)\, p_{0|t}(x_0 \mid x_t, c)$ on each side:

$$w(x_0)\, p_t^{old}(x_t \mid c)\, \pi_{0|t}^{old}(x_0 \mid x_t, c) = Z(c)\, p_t^{\text{ft}}(x_t \mid c)\, \pi_{0|t}^{+}(x_0 \mid x_t, c).$$

Dividing both sides by $p_t^{old}(x_t \mid c)$ and using $\dfrac{p_t^{\text{ft}}(x_t \mid c)}{p_t^{old}(x_t \mid c)} = \dfrac{h(x_t,t,c)}{Z(c)}$ in Equation 2.1

$$w(x_0)\, \pi_{0|t}^{old}(x_0 \mid x_t, c) = Z(c) \cdot \frac{h(x_t,t,c)}{Z(c)} \cdot \pi_{0|t}^{+}(x_0 \mid x_t, c) = h(x_t,t,c)\, \pi_{0|t}^{+}(x_0 \mid x_t, c). \tag{C.3}$$

The same argument with w replaced by $1-w$ (equivalently h replaced by $1-h$, π^+ by π^-) gives $(1 - w(x_0))\, \pi_{0|t}^{old} = (1 - h(x_t,t,c))\, \pi_{0|t}^{-}$.

Reduction to a weighted regression at fixed x_t. In NFT, loss function is defined as

$$\mathcal{L}(\theta, c) = \mathbb{E}_{\pi^{old}(x_0,x_t|c)}[w(x_0)\|v_\theta^+(x_t,t,c) - v\|_2^2 + (1 - w(x_0))\|v_\theta^-(x_t,t,c) - v\|_2^2]$$

Since the joint density $\pi^{old}(x_0, x_t \mid c)$ factorizes as $\pi_t^{old}(x_t \mid c)\, \pi_{0|t}^{old}(x_0 \mid x_t, c)$, the inner expectation of the positive term at fixed (x_t, t, c) is

$$\begin{aligned}\int w(x_0)\, \pi_{0|t}^{old}(x_0 \mid x_t, c)\, \|v_\theta^+ - v\|^2\, dx_0 &= h(x_t,t,c) \int \pi_{0|t}^{+}(x_0 \mid x_t, c)\, \|v_\theta^+ - v\|^2\, dx_0 \\ &= h\, \mathbb{E}_{\pi_{0|t}^+}\big[\|v_\theta^+ - v\|^2\big],\end{aligned}$$

using the identity C.3 and v_θ^+ being independent of x_0. Since v_θ^+ is a constant w.r.t. x_0,

$$\mathbb{E}_{\pi_{0|t}^+}\big[\|v_\theta^+ - v\|^2\big] = \big\|v_\theta^+ - \mathbb{E}_{\pi_{0|t}^+}[v]\big\|^2 + \underbrace{Var_{\pi_{0|t}^+}(v)}_{\text{independent of } \theta} = \|v_\theta^+ - v^+\|^2 + C_+,$$

where $\mathbb{E}_{\pi_{0|t}^+}[v] = v^+(x_t, c, t)$ by definition of the velocity field. The negative term reduces identically to $(1-h)\big[\|v_\theta^- - v^-\|^2 + C_-\big]$. Dropping the θ-independent constants, the loss at fixed x_t, t, c is equivalent to minimizing

$$h\, \|v_\theta^+ - v^+\|_2^2 + (1-h)\, \|v_\theta^- - v^-\|_2^2.$$

Let $Y := v_\theta - v^{old}$. Let $D := v^+ - v^{old}$ and $D^- := v^- - v^{old}$. Substituting the parameterization,

$$v_\theta^+ - v^+ = (1-\beta)v^{old} + \beta v_\theta - (v^{old} + D) = \beta(v_\theta - v^{old}) - D = \beta\Big(Y - \tfrac{D}{\beta}\Big),$$

$$v_\theta^- - v^- = (1+\beta)v^{old} - \beta v_\theta - (v^{old} + D^-) = -\beta(v_\theta - v^{old}) - D^- = -\beta\Big(Y + \tfrac{D^-}{\beta}\Big).$$

Hence $\|v_\theta^+ - v^+\|^2 = \beta^2\|Y - D/\beta\|^2$ and $\|v_\theta^- - v^-\|^2 = \beta^2\|Y + D^-/\beta\|^2$; factoring out $\beta^2 > 0$ (which does not affect the minimizer) gives the weighted least-squares problem

$$f(Y) = h\Big\|Y - \tfrac{D}{\beta}\Big\|_2^2 + (1-h)\Big\|Y + \tfrac{D^-}{\beta}\Big\|_2^2.$$

Solving for the minimizer. f is a weighted sum of squared distances to two points, D/β (weight h) and $-D^-/\beta$ (weight $1-h$); its minimizer is the weighted average

$$Y^* = h \cdot \frac{D}{\beta} + (1-h) \cdot \Big(- \frac{D^-}{\beta} \Big) = \frac{hD - (1-h)D^-}{\beta}.$$

Since $v^{old} = h\, v^+ + (1-h)\, v^-$, subtracting v^{old} from both sides gives $0 = hD + (1-h)D^-$, i.e. $hD = -(1-h)D^-$. Substituting,

$$Y^* = \frac{hD - (1-h)D^-}{\beta} = \frac{hD + hD}{\beta} = \frac{2hD}{\beta}$$

$$\implies \quad v_{\theta^*} = v^{old} - \frac{2h(x_t, t, c)}{\beta} \cdot \frac{t}{1-t} \nabla_x \log h(x_t, t, c).$$

For fixed scalar β, this is a state-dependent damped version of v^{ft}: the guidance strength scales with the local optimality probability h, vanishing where $h \to 0$.

Therefore our algorithm is choosing β in the Definition C.2 to be $2h(x_t, t, c)$. This is reasonable because when you have $(x_t, t \mid c)$ is more likely to be optimal, that is $h(x_t, t, c) > 0.5$, we are more confident to push fine-tuned sampling strategy v_θ to be farther on the current improvement direction. On the other hand if $h(x_t, t, c) < 0.5$, we are not willing to trust v_θ that much and keep closer to v^{old}. However, in DiffusionNFT, they don't estimate h-function explicitly, so choosing β as a function depending on (x_t, t, c) is not viable. They have to leave β as a hyper-parameter (in their implementation, $\beta = 1$ is default value). Our algorithm provided more flexibility on this aspect.

C.4 Single-Branch Optimal Velocity

Training on the positive branch alone,

$$\mathcal{L}^+(\theta) = \mathbb{E}_{c,t,\pi^{old}(x_0|c)} \big[w \, \| v_\theta^+ - v \|_2^2 \big],$$

and using $w\, \pi_{0|t}^{old} = h\, \pi_{0|t}^+$,

$$\mathcal{L}^+(\theta) = \mathbb{E}_{c,t,x_t} \Big[h(x_t, t, c) \cdot \mathbb{E}_{x_0 \sim \pi_{0|t}^+} \| v_\theta^+ - v \|_2^2 \Big].$$

Pointwise in x_t, the L^2-minimizer is the conditional mean regardless of the (positive) weight h:

$$v_\theta^{+*} = \mathbb{E}_{\pi_{0|t}^+}[v] = v^+ = v^{\text{ft}} \quad \implies \quad v_\theta^* = v^{\text{base}} + \frac{D}{\beta} = v^{\text{base}} - \frac{1}{\beta} \cdot \frac{t}{1-t} \nabla_x \log h,$$

which is *exact*, with no h-dependent damping, for any fixed β.

C.5 Unifying the Three via h functions

Comparing the two population optima:

- double-branch: $v_{\theta^*} = v^{\text{base}} - \frac{2h}{\beta} \cdot \frac{t}{1-t} \nabla \log h$,
- single-branch: $v_{\theta^*} = v^{\text{base}} - \frac{1}{\beta} \cdot \frac{t}{1-t} \nabla \log h$.

Setting $\beta(x_t, t, c) = 2h(x_t, t, c)$ in the double-branch objective exactly cancels the damping factor, giving $v_{\theta^*} = v^{\text{ft}}$; equivalently, this coincides with the single-branch optimum at $\beta = 1$. Thus

$$\text{double-branch, } \beta = 2h \quad \Longleftrightarrow \quad \text{single-branch, } \beta = 1 \quad \Longleftrightarrow \quad v_\theta^* = v^{\text{ft}} = v^{\text{base}} - \frac{t}{1-t} \nabla_x \log h.$$

D Discussion on Contingency Table

D.1 Birthday-Deathday Data

The following Table 4 is taken from the classical contingency-table example of (Diaconis & Sturmfels, 1998).

Birth \ Death	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Jan	1	0	0	0	1	2	0	0	1	0	1	0	6
Feb	1	0	0	1	0	0	0	0	0	1	0	2	5
Mar	1	0	0	0	2	1	0	0	0	0	0	1	5
Apr	3	0	2	0	0	0	1	0	1	3	1	1	12
May	2	1	1	1	1	1	1	1	1	1	1	0	12
Jun	2	0	0	0	1	0	0	0	0	0	0	0	3
Jul	2	0	2	1	0	0	0	0	1	1	1	2	10
Aug	0	0	0	3	0	0	1	0	0	1	0	2	7
Sep	0	0	0	1	1	0	0	0	0	0	1	0	3
Oct	1	1	0	2	0	0	1	0	0	1	1	0	7
Nov	0	1	1	1	2	0	0	2	0	1	1	0	9
Dec	0	1	1	0	0	0	1	0	0	0	0	0	3
Total	13	4	7	10	8	4	5	3	4	9	7	8	82

Table 4: Birthday and death month of 82 descendants of Queen Victoria.

D.2 Pretraining the Contingency-Table Flow Model

To obtain the pretrained generative model used in the contingency-table experiment, we first construct a training distribution over 12×12 nonnegative tables with a fixed total count $N = 82$. For each training sample, we randomly generate an integer table $X \in \mathbb{Z}_+^{12\times 12}$ satisfying

$$\sum_{i,j} X_{ij} = N.$$

We then normalize the table by its total count,

$$P_{ij} = \frac{X_{ij}}{N}, \qquad \sum_{i,j} P_{ij} = 1,$$

so that each table can be viewed as a probability vector on the 144-dimensional simplex. A continuous flow model is pretrained on these normalized probability tables.

At inference time, the generated probability table $\widehat{P}$ is converted back to an integer contingency table with total count exactly N. We first compute

$$Y_{ij} = N\widehat{P}_{ij}, \qquad \widetilde{X}_{ij} = \lfloor Y_{ij} \rfloor.$$

Since rounding down may leave a residual

$$R = N - \sum_{i,j} \widetilde{X}_{ij},$$

we assign one additional count to the R entries having the largest fractional parts

$$Y_{ij} - \lfloor Y_{ij} \rfloor.$$

This largest-remainder rounding procedure guarantees

$$\sum_{i,j} \widetilde{X}_{ij} = N$$

while keeping the resulting integer table as close as possible to the continuous model output under this rounding scheme. If the total number is not hard-constrained to be the total number, we could let

$$Y_{ij} = N\widehat{P}_{ij}, \qquad \widetilde{X}_{ij} = \textbf{round}(Y_{ij}).$$

E Additional Details on Text-to-Image Task

ODE Integration. Roll-out time steps are 40 for both training and inference. Note that we only solve the flow ODE step-by-step till $\nabla \log h$, then use one step estimation no matter using Estimator 3.2 or B.1. However, the solver used in training roll-outs is Euler, and during inference is DPM-Solver++(Lu et al., 2025), and CFG scale for online h-guidance w/ CFG is 4 by default, for SD3.5-M is 3.5 by default (both of them are taken due to the relatively best performance among different metrics).

Online Data Collection. Slow mixed policy is used to generate online samples. Follow the default design used in DiffusionNFT(Zheng et al., 2026), $\eta_k = \min(0.001 \cdot k, 0.5)$, which means the data collection starts from on-policy, and gradually converges to $0.5 \cdot v^{old} + 0.5 \cdot v_\theta$ after 500 iterations. On the other hand, for the off policy guidance baseline, the roll-out policy and regression target are based on the base model v^{base}.

Other Configurations. In our T2I implementation, 7 Nvidia L40S is used. For each iteration per GPU, 4 prompts are randomly picked, and 12 initial noise is generated for each prompt. We do not use all time steps in training because two denominators in Flow-equivalent SDE 3.1 will approach to zero when $t \to 1$, therefore we set $t_{\min} = 0.05$ and $t_{\max} = 0.9$. Moreover, stochasticity parameter $\lambda_t = 1$ in SDE 3.1. Learning Rate is $3e-5$, Gradient Clip is set to be 1.0, and slight LoRA regularization $1e-5$ is applied.

F Additional Qualitative Comparison

In this section, we will provide more images showing the aesthetic quality and semantic alignment provided by our algorithm. For baselines, SD3.5-Medium, DiffusionNFT, FlowGRPO and off-policy h-guidance are compared in images.

For aesthetic quality, refer to Figure 3. For semantic alignment, refer to Figure 4.



Figure 3: Qualitative comparison across different methods and prompts. The prompts are: (Prompt 1) “Pixel art of a space station launching with two suns and a comet, featuring a retro space helmet, minigun, and a bright explosion in a dream-like atmosphere with CGA shading.”; (Prompt 2) “The devil using a heat press.”; (Prompt 3) “Ink drawing of a Shaolin monk in fighting pose.”; (Prompt 4) “A side view of a fox, with a flat design, Artwork of t-shirt graphic.”

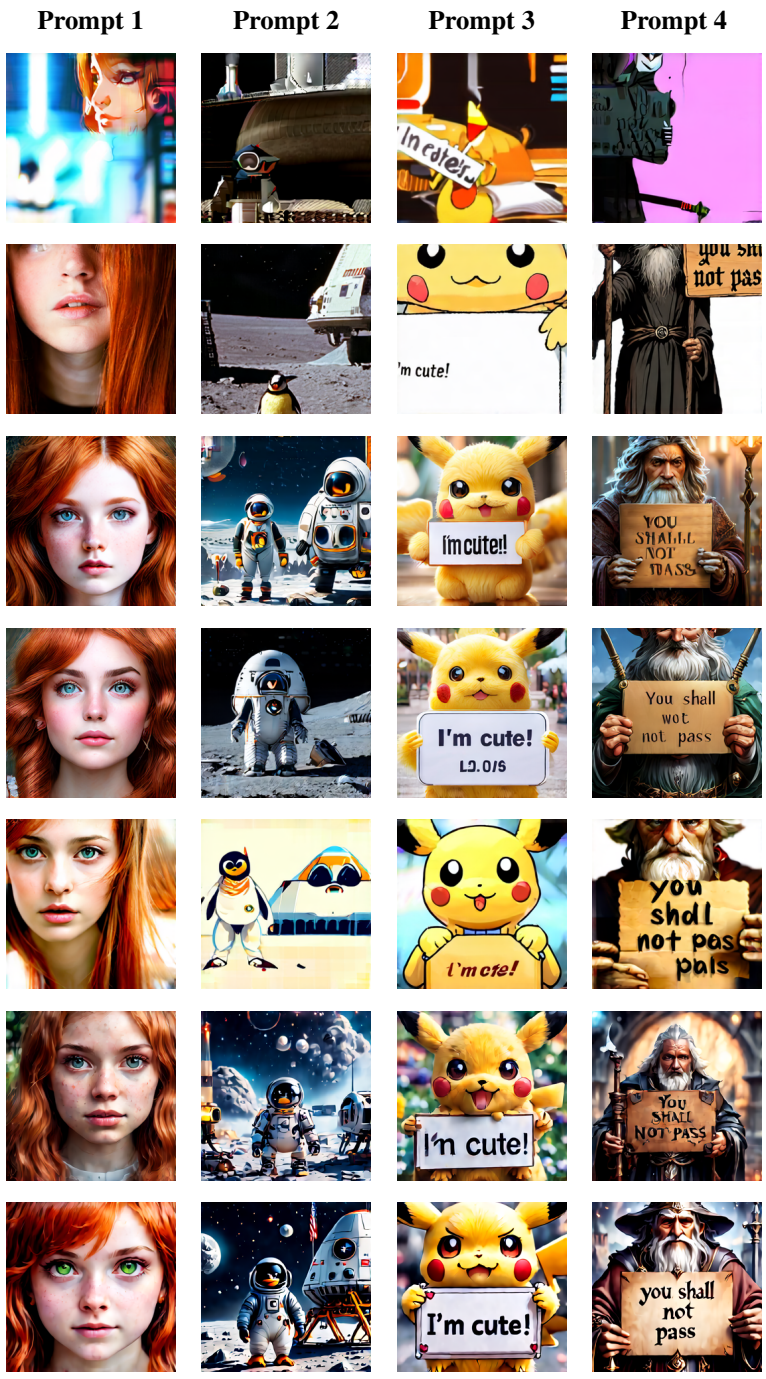

Figure 4: Qualitative comparison across different methods and prompts. The prompts are: (Prompt 1, color) "A young woman with red hairs and green eyes"; (Prompt 2, spatial) "A penguin astronaut, in a penguin spacesuit, standing next to the apollo landing"; (Prompt 3, text) "A pikachu holding a sign that says **"I'm cute!"**"; (Prompt 4, text) "Gandalf holding a sign with a text **"you shall not pass"**".