FENCHEL TILTING: WEIGHTED CORRECTION FOR EFFICIENT FINETUNING OF GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting a pretrained generative model to an arbitrary preference expressed as a utility function underlies reward alignment, guided design, and constraint satisfaction, enabling diverse applications. Existing fine-tuning methods trade off generality against computational cost: they either restrict the family class of supported preferences to keep optimization simple or preserve generality at the expense of efficiency. We introduce Fenchel Tilt Flow Control (FTFC), which decouples utility optimization from generative-model fitting. FTFC first optimizes for a target distribution by jointly fitting an effective reward and density-ratio weights on pretrained samples. Method combines the utility's variational structure with Fenchel duality, supporting general f-divergence penalties that determine how rewards are transformed into an distribution-correction weights. These weights are then frozen and used to modify a diffusion or flow model in a single stage of importance-weighted denoising or flow matching, without differentiating through sampling trajectories. We establish exact duality for concave utilities under suitable conditions and show that weighted fitting reproduces the optimal target distribution for a given utility. Across image and molecule generation benchmarks, FTFC improves over baselines on diverse preference functions, while also being up to $20\times$ more efficient. Proposed method enables adaptation beyond expected-reward maximization without complex optimization, while preserving robustness for more general class of the utility functions compared to baselines.

1 INTRODUCTION

Diffusion and flow models support text-to-image synthesis (Rombach et al., 2022; Esser et al., 2024), molecular generation and drug design (Dunn & Koes, 2024; Schneuing et al., 2024), and robotic control (Chi et al., 2023). Reproducing the training distribution, however, does not directly optimize qualities such as aesthetic appeal, prompt fidelity, or molecular validity. Reward-based fine-tuning adapts a pretrained model to such external scores (Xu et al., 2023), but average reward cannot express all preferences: molecular discovery may prioritize rare high-quality candidates, whereas reliable generation may emphasize the worst outcomes. This motivates optimizing a utility of the full output distribution while remaining close to the pretrained model (De Santi et al., 2025; Wang et al., 2026). For nonlinear utilities, the functional gradient depends on the generated distribution itself (De Santi et al., 2025). Flow Density Control (FDC) handles this dependence by alternating utility linearization and generative fine-tuning, using solvers such as Adjoint Matching (Domingo i Enrich et al., 2025); it therefore requires an outer mirror-descent loop and an inner control solver. Tail-Aware Flow Fine-Tuning (TFFT) separates risk-sensitive threshold selection from a single entropy-regularized update (Wang et al., 2026), but its decomposition is specific to CVaR under KL regularization. Broader structured utilities and divergences require the effective pseudo-reward and induced target distribution to be determined jointly.

Motivated by distribution correction estimation (DICE) in off-policy reinforcement learning (Nachum et al., 2019; Lee et al., 2021), we introduce *Fenchel Tilt Flow Control* (FTFC). FTFC first selects a target distribution relative to the pretrained model and then fits the generator. For concave utilities with suitable f-divergence penalties, Fenchel duality yields a supporting reward and closed-form density-ratio weights that specify the probability-mass shift. The frozen weights then reweight one stage of native denoising or flow-matching training (Zhang et al., 2025; Potaptchik et al., 2025). Precomputed endpoints and rewards, combined with fresh native noise, avoid reward gradients and

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.

differentiation through sampling trajectories. FTFC thus separates *selecting the endpoint distribution* from *learning the generative map*.

Our contributions are:

- **Dual formalization of general utility.** We cast utility optimization as normalized density-ratio fitting with supported f-divergence penalties, including KL and Pearson χ^2. For concave utilities, the Fenchel dual jointly recovers a marginal reward and optimal weights; we also give necessary primal–dual gap conditions.
- **Simple and scalable fine-tuning.** FTFC reduces utility-based adaptation to two simple steps: optimize the target weights on cached samples, then freeze them and fine-tune the generator by reweighting its native denoising or flow-matching loss. No reward gradients, trajectory differentiation, or specialized generative optimization is required.
- **Efficient adaptation.** Across molecular and image benchmarks, FTFC is 10–20$\times$ faster than prior fine-tuning methods. By separating target selection from generator fitting, it avoids repeated reward optimization and differentiation through generation trajectories.

2 Related Work

General Utility Optimization. Flow Density Control (FDC) optimizes nonlinear utilities by alternating utility linearization with entropy-regularized fine-tuning (De Santi et al., 2025), implemented through Adjoint Matching (Domingo i Enrich et al., 2025). Tail-aware Flow Fine-Tuning (TFFT) instead selects a CVaR threshold before one KL-regularized fine-tuning stage (Wang et al., 2026). FTFC extends this separation to structured utilities and supported f-divergences: it determines the marginal reward and target weights before updating the generator. Under KL-regularized CVaR, it recovers the TFFT target objective.

Distribution Correction Estimation (DICE). DICE methods address off-policy distribution shift by estimating state-action occupancy ratios rather than trajectory importance weights. DualDICE estimates these ratios for evaluation (Nachum et al., 2019), whereas OptiDICE uses Fenchel duality to optimize a divergence-regularized occupancy measure from offline data (Lee et al., 2021). FTFC transfers this view to generative fine-tuning by estimating target-to-pretrained density ratios. For nonlinear utilities, the effective reward depends on the unknown target distribution; Fenchel duality resolves this coupling by jointly recovering the marginal reward and normalized weights, with endpoint normalization replacing RL occupancy-flow constraints.

Learning the target generator. A specified target distribution can be learned through stochastic control, including non-KL reward tilts (Tang & Zhou, 2024), or weighted flow and diffusion regression (Zhang et al., 2025). DiffCon derives general-f weighted denoising from a trajectory-distribution objective and an equivalent KL transition-control form (Yang et al., 2026). Tilt Matching avoids reward gradients and trajectory backpropagation (Potaptchik et al., 2025), while online reward-weighted flow matching combines regression, fresh samples, and Wasserstein regularization (Fan et al., 2025). Guidance offers another route (Feng et al., 2025); Diffusion-DICE learns it from occupancy ratios (Mao et al., 2024), and FTFC uses its in-sample estimator only for optional guidance. Its main algorithm instead fits the calibrated endpoint distribution with the native denoising or flow-matching loss while keeping utility-dependent weights fixed.

3 Problem Setup

We adopt the notation of Flow Density Control (FDC) (De Santi et al., 2025) and Tail-aware Flow Fine-Tuning (TFFT) (Wang et al., 2026). We will formalize the problem setting in the deterministic flow-based framework.

Generative models as policies. Let $\mathcal{P}(\mathcal{X})$ denote the Borel probability measures on the sample space $\mathcal{X} = \mathbb{R}^d$. A flow model is specified by a parametrized velocity field $u_\theta(x,t)$: given the current sample x and generation time t, it predicts the direction and speed of the sample's motion, i.e *vector field*. Pretraining learns θ so that transporting initial noise $X_0 \sim p_0$ produces samples approximating the data distribution p_{data} (Lipman et al., 2023; 2024). We interpret (x,t) as a state and the predicted velocity as an action. The generative network itself therefore defines a deterministic *policy* $\pi_\theta(x,t) := u_\theta(x,t)$, abbreviated as π. Sampling follows $X_t = \psi_t^\pi(X_0)$ with

$$\frac{d}{dt}\psi_t^\pi(x) = \pi(\psi_t^\pi(x), t)\,, \qquad \psi_0^\pi(x) = x, \qquad t \in [0,1]. \tag{1}$$

We denote the distribution of X_t by p_t^π, with p_1^π be an endpoint (final) distribution. The pretrained weights θ_{pre} define the fixed reference output law $q := p_1^{\text{pre}}$. Thus p_0 describes input noise, while q describes generated samples.

Generative optimization. Fine-tuning updates the network pretrained parameters θ, changing its movement rule π_θ and hence its output distribution $p_1^{\pi_\theta}$. Given a general (possibly, non-linear) utility function $\mathcal{F} : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$, a divergence $\mathcal{D}$ measuring departure from q, and $\alpha \geq 0$, the objective is

General Utility

$$\sup_{\pi \in \Pi} \mathcal{G}(p_1^\pi), \qquad \mathcal{G}(p) := \mathcal{F}(p) - \alpha \mathcal{D}(p\|q). \tag{2}$$

For a flow backbone, $\Pi = \{\pi_\theta : \theta \in \Theta\}$ is the family of velocity fields represented by the model, with allowed parameter set Θ and fixed source p_0. For a reward $r : \mathcal{X} \to \mathbb{R}$, choosing $\mathcal{F}(p) = \mathbb{E}_{X\sim p}[r(X)]$ and $\mathcal{D} = \mathrm{D}_{\mathrm{KL}}$ recovers standard reward fine-tuning (mean maximizing). TFFT (Wang et al., 2026) instead uses left- or right-CVaR of $r(X)$, with quantile level $\beta \in (0, 1)$, to optimize the lower or upper reward tail by reformulating CVaR objective with an equivalent variational form.

Endpoint distribution correction. The distribution formulation extends to an *f-divergence*, $\mathcal{D} = D_f$. If $p \ll q$, its density ratio $w = dp/dq$ gives $p = wq$, meaning $p(A) = \int_A w(x)q(dx)$, and $D_f(p\|q) = \mathbb{E}_q[f(w)]$ for a convex generator f with $f(1) = 0$. KL is the particular choice $f(s) = s \log s$, with $0 \log 0 = 0$:

$$\mathrm{D}_{\mathrm{KL}}(p\|q) = \mathbb{E}_p[\log w] = \mathbb{E}_q[w \log w].$$

For this formulation, restrict Π to policies with $p_1^\pi \ll q$ and finite objective terms. Write $w_\pi := dp_1^\pi/dq$ and define $\mathcal{W}_\Pi := \{w_\pi : \pi \in \Pi\}$, the set of ratios actually produced by these policies. The objective depends on π only through its terminal law: $p_1^\pi = w_\pi q$ implies $\mathcal{G}(p_1^\pi) = \mathcal{F}(w_\pi q) - \alpha \mathbb{E}_q[f(w_\pi(x))]$. Conversely, every $w \in \mathcal{W}_\Pi$ comes from at least one admissible policy, by definition. Both parameterizations therefore attain exactly the same objective values. Taking suprema yields

$$\begin{aligned} \sup_{\pi\in\Pi} \mathcal{G}(p_1^\pi) &= \sup_{w\in\mathcal{W}_\Pi} \{\mathcal{F}(wq) - \alpha \mathbb{E}_q[f(w)]\} \\ &\leq \sup_{w\geq 0,\ \mathbb{E}_q[w]=1} \{\mathcal{F}(wq) - \alpha \mathbb{E}_q[f(w)]\}. \end{aligned} \tag{3}$$

Every w_π is nonnegative and satisfies $\mathbb{E}_q[w_\pi] = 1$. The inequality enlarges $\mathcal{W}_\Pi$ to all ratios satisfying these constraints, with finite objective terms; absolute continuity alone does not guarantee that every such ratio is realizable by the model. An optimizer $w^\star$ of this relaxed problem, when it exists, defines the target $p^\star = w^\star q$, which provides a connection between optimal distribution maximizing reward and pretrained model.

4 Method: Fenchel Tilt Flow Control

FTFC optimizes the endpoint distribution before fitting the generator through dual optimization (Figures **??**). We consider the relaxed endpoint problem in (3), with $\alpha > 0$.

4.1 Motivation

Consider two rings with reward $r(x) = 1 - d(x)^2$, where $d(x)$ is the distance to the nearer ring. Since points near either ring receive similar reward, high reward does not require covering both rings or their full support. We optimize the lowest-20% CVaR with a KL penalty relative to the pretrained distribution q, so the objective depends not only on sample quality but also on how probability mass is allocated.

Figure 1 compares FTFC with Implicit Tilt Matching (ITM; Potaptchik et al., 2025) and FDC (De Santi et al., 2025), using the same calibrated CVaR reward. FDC attains higher raw CVaR but misses the right ring, while ITM covers both rings but concentrates on particular arcs. FTFC instead most closely matches the numerical target in sliced-Wasserstein distance. The example exposes the key motivation for FTFC: knowing which samples are rewarding is not enough to determine how probability should be distributed among them. FTFC resolves this ambiguity first, by selecting the target distribution by cheap dual optimization, and only then fits the generator to that fixed target using weighted denoising or flow matching (10). Additional objectives are given in Appendix C.1.

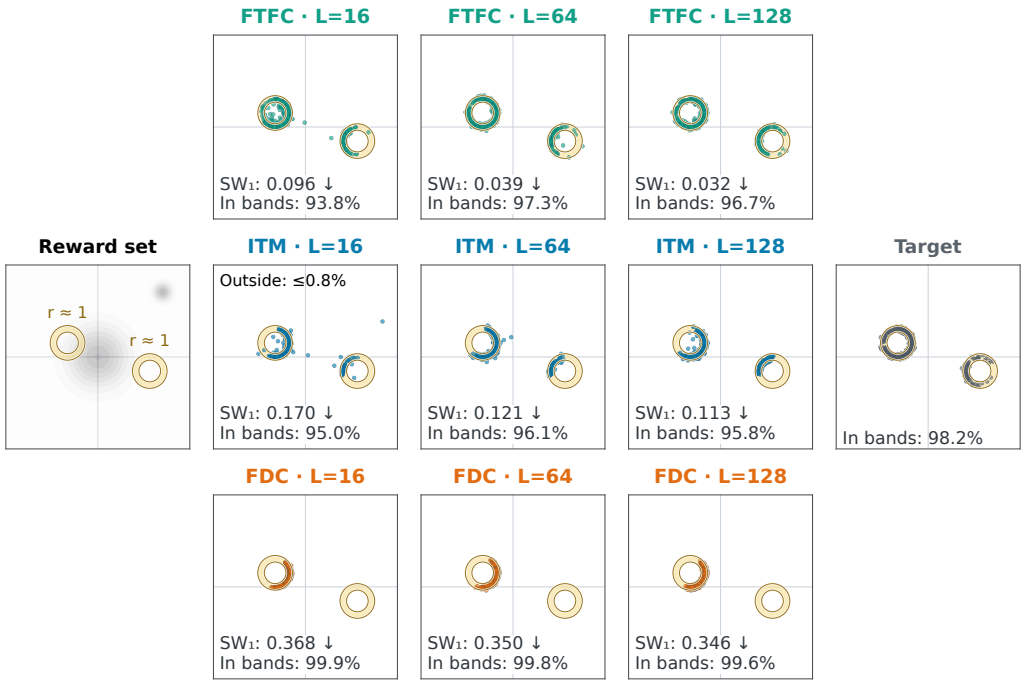

Figure 1: **Left-tail CVaR on two rings.** Gold bands correspond to $d(x) \leq 0.15$. L counts number of sampling steps. SW_1 is distance to the independent numerical target (right, ground truth). FTFC converges faster to target according to the utility function. Comparison is made with FDC (De Santi et al., 2025) and ITM (Potaptchik et al., 2025). See Section 4.1 for details.

4.2 Optimal correction in the closed form

When $\mathcal{F}$ is differentiable, let $g_p : \mathcal{X} \to \mathbb{R}$ denote its *functional derivative at* p (first variation). On a finite sample space, this is $g_p(x_i) = \partial \mathcal{F}(p)/\partial p_i$. The subscript p identifies the distribution being perturbed, and $g_p(x)$ is the derivative at point x. Intuitively, it measures the gain from redistributing probability: moving a small mass from y to x changes utility at rate $g_p(x) - g_p(y)$. To derive the optimal weight, perturb w to $w + \varepsilon v$ in (3), keeping $\mathbb{E}_q[v] = 0$ so the law stays normalized. Since the distribution changes by $\varepsilon v q$, the first variation is

$$\frac{d}{d\varepsilon}\mathcal{G}((w + \varepsilon v)q)\Big|_{\varepsilon=0} = \mathbb{E}_q[(g_{wq} - \alpha f'(w))v]\,.$$

At an interior optimum this vanishes for every admissible such v. Hence the net marginal utility $g_{p^\star} - \alpha f'(w^\star)$ must be a constant $\nu^\star$, q-almost everywhere; otherwise moving mass from a lower value to a higher one would improve the objective. Thus

$$\alpha f'(w^\star(x)) = g_{p^\star}(x) - \nu^\star, \qquad p^\star = w^\star q. \tag{4}$$

Here $\nu^\star$ enforces normalization. For KL, $f(w) = w \log w - w + 1$ has derivative $f'(w) = \log w$. Thus (4) gives $w^\star(x) = \exp((g_{p^\star}(x) - \nu^\star)/\alpha)$; choosing $\nu^\star$ so that $\mathbb{E}_q[w^\star] = 1$ yields

KL Weights

$$\boxed{w^\star(x) = \frac{\exp(g_{p^\star}(x)/\alpha)}{\mathbb{E}_{X\sim q}[\exp(g_{p^\star}(X)/\alpha)]}.} \tag{5}$$

The denominator normalizes the weights so that $\mathbb{E}_q[w^\star] = 1$. Table 4 in Appendix A.1 lists f, f', together with the closed form weight formulas for other divergences. For expected reward, $\mathcal{F}(p) = \mathbb{E}_p[r]$ and $g_p(x) = r(x)$. For nonlinear utility, g_p plays the role of a *marginal reward*: moving probability toward larger values of g_p improves utility to first order, but these values depend on the current distribution. Equation (5) therefore requires the marginal reward at the unknown optimum $p^\star$. We next use dual optimization to determine this reward and the normalized density ratios together.

4.3 FENCHEL DUALITY AND WEIGHT RECOVERY

To find the unknown marginal reward $g_{p^\star}$ from Section 4.2, we now optimize over candidate reward functions $g : \mathcal{X} \to \mathbb{R}$. Here $\mathcal{F}$ is the *same utility functional* as in (2). We assume it is proper, closed, and concave. For each g, define $C(g)$ as the smallest scalar offset satisfying $\mathcal{F}(p) \le C(g) + \mathbb{E}_p[g]$ for every normalized q-supported law p. Equivalently,

$$C(g) := \sup_{p'} \{\mathcal{F}(p') - \mathbb{E}_{p'} g\}, \qquad \mathcal{F}(p) = \inf_{g} \{C(g) + \mathbb{E}_p g\}. \tag{6}$$

At a differentiable p, concavity makes the bound tight for $g = g_p$. More generally, taking the tightest of these bounds recovers $\mathcal{F}$ exactly by concave biconjugacy (Rockafellar, 1970). This representation alone does not find $g_{p^\star}$: we must optimize g and recover its normalized weights.

For any feasible $w \ge 0$, $\mathbb{E}_q w = 1$, candidate reward g, and scalar labeleq:ftfcν with finite terms, the regularized objective satisfies

$$\begin{aligned} \mathcal{G}(wq) &\le C(g) + \mathbb{E}_q[wg - \alpha f(w)] \\ &= C(g) + \nu + \mathbb{E}_q[(g-\nu)w - \alpha f(w)] \\ &\le \underbrace{C(g) + \nu + \alpha \mathbb{E}_q f_+^*((g-\nu)/\alpha)}_{D_f(g,\nu)}. \end{aligned}$$

Appendix A.1 derives the full chain: the first inequality uses the utility bound in (6). The equality uses $\nu(1 - \mathbb{E}_q w) = 0$, introducing the normalization multiplier. The last inequality maximizes each nonnegative weight separately, using $f_+^*(u) := \sup_{v \ge 0}\{uv - f(v)\}$. Minimizing this bound recovers the optimal value of the endpoint relaxation in (3), under the strong-duality conditions in Theorem 1. The weights attaining the last inequality are recovered by conjugacy, with ν chosen to normalize them:

Fenchel Dual

$$\boxed{\begin{aligned} &D_f(g,\nu) := C(g) + \nu + \alpha \mathbb{E}_q f_+^*((g-\nu)/\alpha), \\ &(g^\star, \nu^\star) \in \arg\min_{g,\nu} D_f(g,\nu), \\ &\quad w^\star(x) \in \partial f_+^*((g^\star(x) - \nu^\star)/\alpha), \qquad \mathbb{E}_q w^\star = 1. \end{aligned}} \tag{7}$$

The subgradient allows zero weights. The multiplier ν enforces normalization within the weight response; rescaling the weights afterward is generally invalid beyond KL. For example, half-Pearson uses $f(w) = (w-1)^2/2$ and $w = (1 + (g-\nu)/\alpha)_+$. This conjugate recovery is the DICE connection (Lee et al., 2021), with endpoint normalization replacing occupancy-flow constraints.

Theorem 1 (Duality and optimality gap)**.** *For any feasible $p = wq$ and finite primal and dual terms, put $u = (g - \nu)/\alpha$. Then*

$$\begin{aligned} D_f(g,\nu) - \mathcal{G}(p) = &\underbrace{C(g) + \mathbb{E}_p g - \mathcal{F}(p)}_{\textit{supporting-reward slack}} \\ &+ \underbrace{\alpha \mathbb{E}_q[f(w) + f_+^*(u) - uw]}_{\textit{weight-response residual}} \ge 0. \end{aligned} \tag{8}$$

Thus the gap bounds any improvement in $\mathcal{G}$ achievable by another feasible endpoint law. Under strong duality and attainment, both residuals vanish at an optimum and (7) recovers its weights. A sufficient setting is a finite bank with positive reference masses, finite continuous concave utility, and f finite continuous and strictly convex on $[0,\infty)$, differentiable on $(0,\infty)$, and superlinear.

At zero gap, $C(g^\star) + \mathbb{E}_{p^\star} g^\star = \mathcal{F}(p^\star)$: the affine upper bound in (6) touches $\mathcal{F}$ at the recovered law $p^\star = w^\star q$. At a differentiable interior optimum, their first variations agree, identifying $g^\star$ with $g_{p^\star}$ up to an additive constant. Thus (7) determines the reward and law together. Appendix A.1 proves the result.

4.4 TARGET OPTIMIZATION FOR STRUCTURED UTILITIES

The functional dual in (7) optimizes over an arbitrary reward function g, which is generally impractical. For many structured utilities, however, the unknown effective reward has only a small number of parameters. We assume and exploit this structure to optimize the target weights on the cached sample bank before fitting the generator.

$$\mathcal{F} \xrightarrow[\text{Eq. (7)}]{\min_{g,\nu} D_f(g,\nu)} (g^\star, \nu^\star) \xrightarrow[\text{Eq. (7)}]{w^\star \in \partial f_+^*((g^\star - \nu^\star)/\alpha)} p^\star = w^\star q \xrightarrow[\text{Eq. (10)}]{\min_\theta \mathcal{L}(\theta; w^\star)} \widehat{\pi}$$

Figure 2: **FTFC: select a target, then fit it.** The dual determines $(g^\star, \nu^\star)$; the DICE link recovers normalized endpoint weights, which are frozen for native fitting.

Feature parametrized utilities. Consider $\mathcal{F}(p) = \mathbb{E}_p[r] - \Psi(m_p)$ with $m_p = \mathbb{E}_p[\phi]$ where $\phi : \mathcal{X} \to \mathbb{R}^k$ contains the statistics relevant to the utility and Ψ is proper, closed, and convex. Its marginal reward is

$$g_p(x) = r(x) - \nabla\Psi(m_p)^\top \phi(x),$$

so instead of optimizing an unrestricted g, it suffices to determine the k marginal costs $z = \nabla\Psi(m_p)$. Using $\Psi^*(z) = \sup_m\{z^\top m - \Psi(m)\}$ gives

Feature Dual

$$\begin{aligned} \mathcal{F}(p) &= \inf_z\{\Psi^*(z) + \mathbb{E}_p[r - z^\top\phi]\}, \\ (z^\star, \nu^\star) &\in \arg\min_{z,\nu}\{\Psi^*(z) + \nu + \alpha\mathbb{E}_q f_+^*((r - z^\top\phi - \nu)/\alpha)\}, \\ g^\star &= r - z^{\star\top}\phi, \qquad z^\star \in \partial\Psi(\mathbb{E}_{p^\star}\phi). \end{aligned} \tag{9}$$

Thus target optimization reduces to solving for $(z^\star, \nu^\star)$ on cached samples. The resulting $g^\star$ is converted into normalized target weights by (7). Representative utilities and their corresponding (ϕ, Ψ) are given in Appendix A.1; numerical details are given in Appendix A.2.

CVaR utilities. CVaR does not require moment features: its effective reward is determined by a scalar threshold. Upper-CVaR follows the concave formulation above with scalar threshold optimization. Lower-CVaR is convex in the output law and therefore requires a separate argument. On a finite sample bank, we show that a globally optimal lower-CVaR threshold can be chosen among the observed rewards (Proposition 3). Searching these thresholds yields the target weights, which are then frozen for the same generator-fitting stage. Benchmark-specific pseudo-rewards and threshold objectives are given in Appendix A.2.

4.5 Weighted denoising and flow matching

Calibration in Section 4.4 returns weights w defining the target $p = wq$. Freeze these weights and initialize the generator from its pretrained parameters. The calibrated reward affects the generator through the weighted training loss. For each endpoint $X \sim q$, independently sample a time $t \sim \mathcal{U}(0,1)$ and noise ϵ. Weight the usual denoising or flow-matching loss by $w(X)$. Since $p = wq$, this is equivalent to training on the target distribution:

$$\begin{aligned} \mathcal{L}(\theta; w) &= \mathbb{E}_{X\sim q,\, t,\epsilon}[w(X)\ell_\theta(X,t,\epsilon)] \\ &= \mathbb{E}_{X\sim p,\, t,\epsilon}[\ell_\theta(X,t,\epsilon)]. \end{aligned} \tag{10}$$

Here ℓ_θ is the standard denoising or flow-matching loss (Zhang et al., 2025):

$$\begin{aligned} \ell_\theta(X,t,\epsilon) &= \|m_\theta(Y_t,t) - A_t\|^2, \\ (Y_t, A_t) &= \begin{cases} (a_t X + \sigma_t\epsilon,\ \epsilon), & \text{diffusion}, \quad \epsilon \sim \mathcal{N}(0,I), \\ ((1-t)\epsilon + tX,\ X - \epsilon), & \text{flow}, \quad \epsilon \sim p_0. \end{cases} \end{aligned} \tag{11}$$

The respective optimal predictors are the target noise predictor and velocity field, since conditional regression gives

$$m_p(y,t) = \mathbb{E}_p[A_t \mid Y_t = y] = \frac{\mathbb{E}_q[w(X)A_t \mid Y_t = y]}{\mathbb{E}_q[w(X) \mid Y_t = y]}. \tag{12}$$

These identities require finite conditional moments, $\sigma_t > 0$, and a positive denominator. Fresh noise preserves the chosen source law; reweighting cached generation trajectories generally does not.

Equation (12) justifies the weighted regression used by FTFC. For diffusion models, the same calibrated endpoint distribution also admits an alternative realization through score guidance, which we detail below.

Algorithm 1 Fenchel Tilt Flow Control (FTFC)

Require: Pretrained model θ_0 with endpoint distribution q; utility $\mathcal{F}_N$; divergence f; regularization α; sample bank size N; training steps K

Sample and cache endpoints $\{x_i\}_{i=1}^N \sim q$ and evaluate their rewards and utility features

Stage 1: Calibrate the target distribution

Compute the supporting rewards $\{\widehat{g}_i\}_{i=1}^N$ from the utility ▷ Sec. 4.4

Compute normalized density-ratio weights ▷ Eq. (7)

$$\widehat{w}_i \in \partial f_+^*\left(\frac{\widehat{g}_i - \widehat{\nu}_{h_i}}{\alpha}\right), \qquad \sum_{i \in I_h} a_i \widehat{w}_i = \rho_h$$

Freeze $\widehat{w}$

Stage 2: Fit the generator

$\theta \leftarrow \theta_0$

for $k = 1, \ldots, K$ **do**

  Sample a minibatch of cached endpoints $x_i \sim q$

  Sample fresh t and noise ϵ, and construct the native input-target pair (Y_t, A_t) as in (11) ▷ Eq. (11)

  $L(\theta) \leftarrow \frac{1}{B} \sum_{i \in \mathcal{B}} \widehat{w}_i \| m_\theta(Y_t^{(i)}, t) - A_t^{(i)} \|^2$ ▷ Eq. (10)

  $\theta \leftarrow \theta - \eta_k \nabla_\theta L(\theta)$

return θ

Theorem 2 (Score correction under endpoint reweighting). *Let $w^\star \geq 0$, $\mathbb{E}_q w^\star = 1$ be recovered by (7), and set $p^\star = w^\star q$. For $Y_t = a_t X + \sigma_t \epsilon$ with independent $\epsilon \sim \mathcal{N}(0, I)$ and $\sigma_t > 0$, denote the noisy densities by $\widetilde{q}_t, \widetilde{p}_t^\star$ and define $h_t(y) = \mathbb{E}_q[w^\star(X) \mid Y_t = y]$. Assume $0 < h_t < \infty$, $\mathbb{E}_{t,\widetilde{q}_t} |\log h_t| < \infty$, and differentiation under the integral is valid. Writing $s_q = \nabla_y \log \widetilde{q}_t$ and $\epsilon_q = \mathbb{E}_q[\epsilon \mid Y_t = y]$ (and analogously for $p^\star$), we have*

$$\begin{aligned} \widetilde{p}_t^\star(y) &= h_t(y) \widetilde{q}_t(y), \\ s_{p^\star}(y,t) &= s_q(y,t) + \nabla_y \log h_t(y), \\ \epsilon_{p^\star}(y,t) &= \epsilon_q(y,t) - \sigma_t \nabla_y \log h_t(y). \end{aligned} \tag{13}$$

Over integrable scalar functions, the in-sample loss has the unique minimizer

$$\boxed{\begin{aligned} \mathcal{L}_{\mathrm{IGL}}(\ell; w^\star) &:= \mathbb{E}_{t, X \sim q, \epsilon}[w^\star(X) e^{-\ell(Y_t, t)} + \ell(Y_t, t)], \\ \ell^\star(y,t) &= \log h_t(y), \qquad \nabla_y \ell^\star = s_{p^\star} - s_q. \end{aligned}} \tag{14}$$

The result holds for any normalized nonnegative weights, including general-f responses with zeros. Weighted denoising (10) learns the same target predictor directly; Algorithm 1 uses this route. Appendix A.3 proves the theorem and gives the corresponding correction for Gaussian flow interpolations. Under exact realization of the weighted regression target, the fitted generator recovers the selected endpoint law $p^\star = w^\star q$ and therefore inherits its endpoint optimality; we formalize this result and its regularity conditions in Appendix A.3.

5 Experiments

5.1 Molecular generation on QM9

We adapt FlowMol (Dunn & Koes, 2024) on QM9 (Ramakrishnan et al., 2014) using negative GFN1-xTB energy (Friede et al., 2024), with an upper-superquantile objective targeting the best 0.2% of rewards. We compare FTFC with pretrained FlowMol, expected-reward Adjoint Matching (AM; Domingo i Enrich et al. 2025), FDC (De Santi et al., 2025), and TFFT (Wang et al., 2026) under the shared ODE evaluation in Table 1.

Tail reward and molecular geometry. FTFC attains the highest observed mean and both upper-tail rewards. It also exceeds every fine-tuned baseline on topology reward, which measures connectivity preservation and relaxation strain (Kotani, 2026), while remaining competitive in synthetic accessibility (Ertl & Schuffenhauer, 2009). Thus FTFC offers a better observed reward–geometry trade-off than these fine-tuned alternatives. Including sample-bank acquisition, FTFC takes about one hour per seed, versus $12 - 35$ hours for the baselines.

Table 1: **QM9 / FlowMol molecular generation.** ODE evaluation with 50,000 attempts per seed. Rewards are in Hartree; bold denotes the best observed value.

Method	Train days ↓	Mean R ↑	SQ$_{.998}$ ↑	Upper 10% ↑	Valid. (%) ↑	SA ↓	Topology R_{valid} ↑
Pretrained	0.0	28.6	38.7	34.1	**96.4**	4.6	0.8
AM	0.5	28.4	38.1	33.5	96.4	4.7	0.7
FDC	1.4	28.6	38.4	33.6	95.1	**4.4**	0.6
TFFT	0.6	28.9	39.2	34.4	94.2	4.7	0.7
FTFC	**0.04**	**29.1** ±0.1	**39.4** ±0.3	**35.5** ±0.1	96.1 ±1.8	**4.4** ±0.1	**0.8** ±0.0

These runs show that one fitting stage with fixed endpoint weights can improve a nonlinear tail objective, without repeated online reward optimization. These comparisons use the recorded method budgets, with three FTFC seeds and one run per baseline. Appendix B.2 gives the protocol, metric definitions, and timing scope; it also documents TFFT's inactive tail-reward gradients in this reconstruction.

5.2 Text-to-image generation

We evaluate FTFC on the Stable Diffusion v1.5 benchmark of Wang et al. (2026). ImageReward is the sole fine-tuning reward: EXP-FT optimizes its expectation, while FDC and L-TFFT target lower-tail CVaR at $\beta = 0.2$ with KL regularization. CLIP (Radford et al., 2021), HPSv2.1 (Wu et al., 2023), and DreamSim (Fu et al., 2023) are evaluation-only metrics. We generate ten images for each of 100 released prompts; Table 2 reports the results, and Appendix B.4 gives settings, metric definitions, and uncertainty estimates.

Table 2: **Stable Diffusion v1.5 image generation.** Evaluation over 1,000 images. Values are mean ± standard error across five seeds; bold denotes the best observed mean.

	ImageReward		Alignment		Diversity	Train
Method	Mean ↑	L-CVaR$_{.2}$ ↑	CLIP ↑	HPSv2.1 ↑	DreamSim var. ↑	days ↓
Pretrained	0.3 ±0.1	−1.0 ±0.0	0.28 ±0.00	0.26 ±0.00	0.33 ±0.01	0.0
EXP-FT	0.7 ±0.0	−0.5 ±0.0	0.28 ±0.00	**0.28** ±0.00	0.30 ±0.01	3.4
FDC	0.7 ±0.1	−0.5 ±0.0	0.28 ±0.00	0.27 ±0.00	0.30 ±0.01	3.6
L-TFFT	0.8 ±0.1	−0.5 ±0.0	0.28 ±0.00	0.27 ±0.00	0.30 ±0.01	2.8
FTFC	**0.9** ±0.1	**−0.4** ±0.0	**0.28** ±0.00	0.27 ±0.00	**0.35** ±0.01	**0.2**

Training time is in A100 GPU-days; evaluation details are given in Appendix B.4.

Reward improvement without diversity collapse. FTFC raises mean and lower-tail ImageReward while keeping DreamSim variance at the pretrained level; all fine-tuned baselines reduce it. Thus, the gain is not explained by concentrating probability on a narrow set of high-reward outputs. FTFC instead selects and fits a fixed redistribution of pretrained mass. EXP-FT and L-TFFT improve reward with a larger diversity shift, whereas FTFC obtains this trade-off in one frozen-weight fitting stage.

Qualitative comparison. Figure 3 shows the lowest-, median-, and highest-reward samples for the prompt *footage of an astronaut in a tropical beach*. Fine-tuning improves the worst samples but can alter the output distribution: L-TFFT, for example, favors saturated, fantastical skies despite high ImageReward. FTFC improves low-reward samples while retaining greater visual variation, consistent with its DreamSim diversity. Appendix B.4 provides details.

5.3 Molecular design on GEOM-Drugs

We evaluate FlowMol3 (Dunn & Koes, 2025) on GEOM-Drugs (Axelrod & Gómez-Bombarelli, 2022), independently reconstructing the energy-guided setting of Wang et al. (2026). FDC, R-TFFT, and FTFC target the upper 10% of negative GFN1-xTB rewards; EXP-FT uses AM to maximize expected reward. All methods share the backbone, reward oracle, and three-seed evaluation in Table 3.

Preserving molecular quality during adaptation. FTFC has the highest observed graph validity and topology reward, with energy rewards near pretrained and above EXP-FT and R-TFFT. FDC attains higher mean and tail energy rewards but lowers topology reward through more protocol failures and greater relaxation strain, showing that higher energy reward can accompany worse geometry.



Figure 3: **Qualitative Stable Diffusion v1.5 comparison.** The prompt is "*footage of an astronaut in a tropical beach*". Each method panel shows ranks 1 (lowest), 5, and 10 (highest), corresponding to ImageReward score. See Section 5.2.

FTFC avoids the larger degradation of the fine-tuned baselines; its small edge over pretrained is not statistically established. Figure 4 shows one threshold-selected molecular panel, with further examples and protocol details in Appendix B.3.1.

Table 3: **GEOM-Drugs / FlowMol3 molecular design.** Evaluation over three seeds and 2,000 molecules per seed. Bold denotes the best observed mean.

Method	Mean $R\uparrow$	R-CVaR $_{0.9}\uparrow$	Valid. (%) $\uparrow$	SA $\downarrow$	Topology $R_{\text{valid}}\uparrow$
Pretrained	65.9 ± 0.1	94.9 ± 0.2	99.8 ± 0.0	7.5 ± 0.01	0.95 ± 0.0
EXP-FT	67.9 ± 0.8	94.7 ± 1.5	99.8 ± 0.2	7.3 ± 0.03	0.93 ± 0.0
FDC	69.2 ± 2.4	**97.9** ± 1.9	99.6 ± 0.2	7.2 ± 0.01	0.93 ± 0.0
R-TFFT	70.6 ± 0.8	90.8 ± 0.9	99.8 ± 0.2	7.3 ± 0.03	0.93 ± 0.0
FTFC	**75.3** ± 0.2	95.0 ± 0.3	**99.9** ± 0.1	**7.1** ± 0.01	**0.97** ± 0.0

Mean $\pm$ standard deviation over three seeds. Topology includes failed draws; details in Appendix B.3.

Appendix B.3 contains the protocol, metric definitions, and molecular examples. It also documents the absent positive-β tail-reward gradients in our R-TFFT reconstruction, which limit conclusions about that baseline.

6 Conclusion & Limitations

We introduced Fenchel Tilt Flow Control (FTFC), a distribution-correction approach to utility-based generative fine-tuning. The key idea is to separate *which distribution to generate* from *how to generate it*: FTFC first optimizes normalized density-ratio weights defining the target distribution, then freezes them and fits the generator with its native denoising or flow-matching objective. This enables nonlinear utilities and general f-divergence penalties without repeated reward optimization during generator training. Across image and molecular generation tasks, FTFC achieves strong performance with substantially lower training time.

FTFC remains limited by the coverage of the pretrained distribution and its assumptions on finite sample bank. Moreover, the theoretical target-optimization guarantees do not account for finite-bank estimation or imperfect generator fitting, and the general dual requires suitable concavity and regularity conditions. Extending FTFC beyond fixed sample banks and to broader nonconcave utilities are promising directions.

REPRODUCIBILITY STATEMENT

The Appendix contains the details required to reproduce our results, including dataset and checkpoint information, preprocessing, evaluation protocols, model and optimizer configurations, all hyperparameters, training budgets, random seeds, and metric definitions. It also gives the relevant derivations, background, details on baselines, and implementation-specific settings used in the experiments. Also, we provide a fully reproducible codebase in the supplementary material.

ETHICS STATEMENT

This work uses publicly available datasets and pretrained models and does not involve human subjects, personal data, or human-subject experiments. The method is intended for research on generative modeling.

AI USE STATEMENT

AI tools were used only to polish the manuscript text and improve grammar and readability. All technical content, experiments, and final wording were reviewed and verified by the authors.

REFERENCES

Simon Axelrod and Rafael Gómez-Bombarelli. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9:185, 2022. doi: 10.1038/s41597-022-01288-4.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL https://stanford.edu/~boyd/cvxbook/.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Flow density control: Generative optimization beyond entropy-regularized fine-tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2511.22640.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025.

Ian Dunn and David R. Koes. Mixed continuous and categorical flow matching for 3D de novo molecule generation. *arXiv preprint arXiv:2404.19739*, 2024. URL https://arxiv.org/abs/2404.19739.

Ian Dunn and David R. Koes. FlowMol3: Flow matching for 3D de novo small-molecule generation. *arXiv preprint arXiv:2508.12629*, 2025. URL https://arxiv.org/abs/2508.12629.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1:8, 2009. doi: 10.1186/1758-2946-1-8.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Jiajun Fan, Shuaike Shen, Chaoran Cheng, Yuxin Chen, Chumeng Liang, and Ge Liu. Online reward-weighted fine-tuning of flow matching with Wasserstein regularization. *arXiv preprint arXiv:2502.06061*, 2025. URL https://arxiv.org/abs/2502.06061.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. In *Proceedings of the International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.02150.

Marvin Friede, Christian Hölzer, Sebastian Ehlert, and Stefan Grimme. dxtb—an efficient and fully differentiable framework for extended tight-binding. *The Journal of Chemical Physics*, 161(6): 062501, 2024. doi: 10.1063/5.0216715.

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. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://arxiv.org/abs/2306.09344.

Takao Kotani. Atomic design transformer: Scaffold-conditioned 3d molecule generation via xTB-reward reinforcement learning. *arXiv preprint arXiv:2607.15918v1*, 2026. URL https://arxiv.org/abs/2607.15918v1.

Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6120–6130. PMLR, 2021. URL https://proceedings.mlr.press/v139/lee21f.html.

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

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

Liyuan Mao, Haoran Xu, Xianyuan Zhan, Weinan Zhang, and Amy Zhang. Diffusion-DICE: In-sample diffusion guidance for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b2fea79b1137d917e8b7cce9434ab5fa-Paper-Conference.pdf.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1906.04733.

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

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. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014. doi: 10.1038/sdata.2014.22. URL https://www.nature.com/articles/sdata201422.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970. doi: 10.1515/9781400873173. URL https://www.degruyterbrill.com/document/doi/10.1515/9781400873173/html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L. Blundell, Pietro Liò, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, 2024. doi: 10.1038/s43588-024-00737-x. URL https://doi.org/10.1038/s43588-024-00737-x.

Wenpin Tang and Fuzhong Zhou. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond. *arXiv preprint arXiv:2403.06279*, 2024. URL https://arxiv.org/abs/2403.06279.

Zifan Wang, Riccardo De Santi, Xiaoyu Mo, Michael M. Zavlanos, Andreas Krause, and Karl H. Johansson. Efficient tail-aware generative optimization via flow model fine-tuning. *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.16796.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 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. *arXiv preprint arXiv:2304.05977*, 2023. URL https://arxiv.org/abs/2304.05977.

Tong Yang, Moonkyung Ryu, Chih-Wei Hsu, Guy Tennenholtz, Yuejie Chi, Craig Boutilier, and Bo Dai. Diffusion controller: Framework, algorithms and parameterization. *arXiv preprint arXiv:2603.06981*, 2026. URL https://arxiv.org/abs/2603.06981.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.04975.

A Additional Background Derivations

This appendix derives the utility dual, its finite-bank form, and the transfer from endpoint weights to a generative model.

A.1 Utility duality and weight recovery

From the original objective to the dual bound. Let $p \ll q$ with density ratio $w = dp/dq \geq 0$, $\mathbb{E}_q w = 1$, and assume the displayed terms are finite. Change of measure and the definition of the f-divergence give, respectively,

$$\mathbb{E}_{wq}[g] = \int g(x)\, dp(x) = \int g(x) w(x)\, dq(x) = \mathbb{E}_q[wg],$$
$$D_f(wq\|q) = \int f\left(\frac{dp}{dq}(x)\right) dq(x) = \mathbb{E}_q[f(w)].$$

The supremum defining $C(g)$ includes the choice $p' = wq$, so

$$C(g) = \sup_{p'}\{\mathcal{F}(p') - \mathbb{E}_{p'}[g]\} \geq \mathcal{F}(wq) - \mathbb{E}_{wq}[g].$$

Substituting into the original objective (2) yields

$$\begin{aligned}\mathcal{G}(wq) &= \mathcal{F}(wq) - \alpha D_f(wq\|q)\\ &= \mathcal{F}(wq) - \alpha\mathbb{E}_q[f(w)]\\ &\leq C(g) + \mathbb{E}_{wq}[g] - \alpha\mathbb{E}_q[f(w)]\\ &= C(g) + \mathbb{E}_q[wg] - \alpha\mathbb{E}_q[f(w)]\\ &= C(g) + \mathbb{E}_q[wg - \alpha f(w)].\end{aligned}$$

Next, introduce a scalar multiplier ν for normalization. Since $\mathbb{E}_q w = 1$, adding $\nu(1 - \mathbb{E}_q w) = 0$ leaves the bound unchanged:

$$\begin{aligned}L(w, g, \nu) &:= C(g) + \mathbb{E}_q[wg - \alpha f(w)] + \nu(1 - \mathbb{E}_q w)\\ &= C(g) + \nu + \mathbb{E}_q[(g - \nu)w - \alpha f(w)].\end{aligned}$$

For fixed g and ν, maximize the integrand over a nonnegative scalar weight v at each x. Factoring out $\alpha > 0$ gives

$$\begin{aligned}\sup_{v\geq 0}\{(g(x) - \nu)v - \alpha f(v)\} &= \alpha \sup_{v\geq 0}\left\{\frac{g(x) - \nu}{\alpha} v - f(v)\right\}\\ &= \alpha f_+^*\left(\frac{g(x) - \nu}{\alpha}\right), \qquad f_+^*(u) := \sup_{v\geq 0}\{uv - f(v)\}.\end{aligned}$$

The value at $v = w(x)$ cannot exceed this supremum. Taking expectation under q therefore yields

$$\begin{aligned}\mathcal{G}(wq) &\le L(w, g, \nu) \\ &\le C(g) + \nu + \alpha \mathbb{E}_q f_+^* \left(\frac{g - \nu}{\alpha} \right) =: D_f(g, \nu).\end{aligned}$$

For proper closed convex f, equality in the pointwise bound holds when $w(x) \in \partial f_+^*((g(x) - \nu)/\alpha)$; imposing $\mathbb{E}_q w = 1$ gives the response in (7). The bound holds for every finite (g, ν); equality of optimized primal and dual requires the regularity conditions below.

For the exact population envelope in (6), use the $L^1(q)$–$L^\infty(q)$ pairing and a proper, concave, norm-upper-semicontinuous extension of $w \mapsto \mathcal{F}(wq)$; concave biconjugacy then applies.

Response to a fixed reward. Under strong duality for normalization, the optimized response value is

$$\begin{aligned}T_f(g) &:= \sup_{w \ge 0,\, \mathbb{E}_q w = 1} \{\mathbb{E}_q[wg] - \alpha \mathbb{E}_q f(w)\} \\ &= \inf_\nu \{\nu + \alpha \mathbb{E}_q f_+^*((g - \nu)/\alpha)\}.\end{aligned} \tag{15}$$

Integrated Fenchel equality recovers the normalized weights in (7). Write $u = (g(x) - \nu)/\alpha$. For the strictly convex generators in Table 4, a positive maximizer of $uv - f(v)$ satisfies $f'(v) = u$. Inverting f' on its range gives the interior response; the table also accounts for zero weights and the admissible domain of u.

Table 4: Divergence generators, derivatives for $w > 0$, and scalar weight responses $w(u) = \arg\max_{v \ge 0}\{uv - f(v)\}$, where $u = (g(x) - \nu)/\alpha$ and $(a)_+ = \max\{a, 0\}$.

Divergence	$f(w)$	$f'(w)$	$w(u)$	Domain of u
KL	$w \log w - w + 1$	$\log w$	e^u	$\mathbb{R}$
Half-Pearson	$\frac{1}{2}(w-1)^2$	$w - 1$	$(1 + u)_+$	$\mathbb{R}$
Reverse KL	$-\log w + w - 1$	$1 - 1/w$	$(1 - u)^{-1}$	$u < 1$

Half-Pearson permits $w = 0$ when $u \le -1$. Reverse KL has $f(0) = +\infty$ and requires $u < 1$ for a finite maximizing weight. In every row, ν is chosen so that $\mathbb{E}_q[w] = 1$.

For KL with finite Z_g, $f'(w) = \log w$ gives $w_g(x) = e^{(g(x)-\nu)/\alpha}$. Normalization gives $1 = e^{-\nu/\alpha}\mathbb{E}_q[e^{g/\alpha}]$, hence

$$\begin{aligned}Z_g &= \mathbb{E}_{X \sim q}[e^{g(X)/\alpha}], \qquad \nu_g = \alpha \log Z_g, \\ T_{\mathrm{KL}}(g) &= \alpha \log Z_g, \qquad w_g(x) = e^{g(x)/\alpha}/Z_g.\end{aligned} \tag{16}$$

Taking $g = g_{p^\star}$ yields (5).

Proof of Theorem 1. Gap identity. Put $u = (g - \nu)/\alpha$. Since $\mathbb{E}_q w = 1$,

$$\begin{aligned}D_f(g, \nu) - \mathcal{G}(wq) &= C(g) + \nu + \alpha \mathbb{E}_q f_+^*(u) - \mathcal{F}(wq) + \alpha \mathbb{E}_q f(w) \\ &= \underbrace{C(g) + \mathbb{E}_q[wg] - \mathcal{F}(wq)}_{\ge 0} + \alpha \underbrace{\mathbb{E}_q[f(w) + f_+^*(u) - uw]}_{\ge 0}.\end{aligned}$$

The first term is the envelope gap and the second is the Fenchel–Young gap. Both vanish exactly when the envelope is tight and $u \in \partial(f + I_{[0,\infty)})(w)$ almost surely, proving optimality and recovering the weights by conjugacy.

Finite-bank attainment. Let $a_i > 0$, $b \in \Delta_N$, and $J(b) = \alpha \sum_i a_i f(b_i/a_i)$. Continuity on the compact simplex gives a maximizer b^* of $\mathcal{F}_N - J$; strict convexity of J makes it unique. Write $\ell_0 = f'_+(0)$. If $\ell_0 = -\infty$, this maximizer is interior. Indeed, for $b_\varepsilon = (1 - \varepsilon)b^* + \varepsilon a$,

$$\frac{\mathcal{F}_N(b_\varepsilon) - \mathcal{F}_N(b^*)}{\varepsilon} \ge \mathcal{F}_N(a) - \mathcal{F}_N(b^*), \qquad \frac{J(b_\varepsilon) - J(b^*)}{\varepsilon} \longrightarrow -\infty$$

if any coordinate of b^* is zero, contradicting optimality.

When ℓ_0 is finite, extend f below zero by its tangent at zero. In the steep case, extend it below a positive ratio smaller than all b_i^*/a_i. This yields a finite differentiable convex penalty $\bar{J}$ agreeing with J near b^*. The convex subgradient sum rule gives

$$0 \in \partial(-\mathcal{F}_N + I_{\Delta_N})(b^*) + \nabla \bar{J}(b^*),$$
$$\mathcal{F}_N(b) \le \mathcal{F}_N(b^*) + g^{*\top}(b - b^*), \qquad b \in \Delta_N,$$

where $g_i^* = \alpha f'(b_i^*/a_i)$ for positive ratios and $g_i^* = \alpha \ell_0$ at zero. Thus, with $\nu^* = 0$,

$$C_N(g^*) + g^{*\top} b^* = \mathcal{F}_N(b^*), \qquad g_i^*/\alpha \in \partial(f + I_{[0,\infty)})(b_i^*/a_i).$$

Both residuals vanish, establishing dual attainment and strong duality. The argument also holds on a product of scaled simplices for fixed condition masses. For a barrier such as reverse KL, compactness, lower semicontinuity, and $f(0) = +\infty$ give an interior optimum; the same local argument applies.

KL specialization. With $f(w) = w \log w - w + 1$, $f_+^*(u) = e^u - 1$, and normalized $w_g = e^{g/\alpha}/Z_g$,

$$\mathbb{E}_q[f(w) + f_+^*(\log w_g) - w \log w_g] = \mathbb{E}_q[w \log(w/w_g) - w + w_g] = \mathrm{D}_{\mathrm{KL}}(wq \| w_g q).$$

Consequently,

$$\begin{aligned} D_{\mathrm{KL}}(g) - \mathcal{G}(p) &= C(g) + \mathbb{E}_p g - \mathcal{F}(p) + \alpha \mathrm{D}_{\mathrm{KL}}(p \| w_g q), \\ D_{\mathrm{KL}}(g) &= C(g) + \alpha \log Z_g. \end{aligned} \tag{17}$$

At an optimal supporting reward, $w_g q = p^*$; hence for any finite candidate dual value D,

$$\alpha \mathrm{D}_{\mathrm{KL}}(p \| p^*) \le \mathcal{G}(p^*) - \mathcal{G}(p) \le D - \mathcal{G}(p).$$

□

Sufficient population conditions for the structured dual. Let r, ϕ be bounded, $\Psi : \mathbb{R}^k \to \mathbb{R}$ finite continuous convex, and let f satisfy the finite-bank assumptions of Theorem 1, with $f(1) = 0$. Then (9) has zero duality gap and attained primal and dual optima; the optimal ratio is unique. To see this, define

$$A(w) = \alpha \mathbb{E}_q f(w) - \mathbb{E}_q[wr] + I_{\{w \ge 0,\, \mathbb{E}_q w = 1\}}(w), \qquad Bw = \mathbb{E}_q[w\phi].$$

The integral functional is closed convex, B is continuous, and $A(1) < \infty$. Continuity of Ψ at $B1$ qualifies Fenchel–Rockafellar duality (Rockafellar, 1970):

$$\inf_w \{A(w) + \Psi(Bw)\} = \max_z \{-\Psi^*(z) - T_f(r - z^\top \phi)\}.$$

The value is finite because rewards and attainable moments are bounded and $\mathbb{E}_q f(w) \ge f(1) = 0$. A minimizing sequence has bounded $\mathbb{E}_q f(w)$; superlinearity gives uniform integrability and weak compactness in $L^1(q)$. Weak lower semicontinuity then gives primal attainment, and strict convexity gives uniqueness.

For any bounded score s, the conjugate response $R(u) = \arg\max_{v \ge 0}\{uv - f(v)\}$ is continuous, nondecreasing, with $R(-\infty) = 0$ and $R(+\infty) = +\infty$. Consequently

$$\mathbb{E}_q R((s - \nu)/\alpha) \longrightarrow \begin{cases} +\infty, & \nu \to -\infty, \\ 0, & \nu \to +\infty, \end{cases} \qquad \exists \nu :\ \mathbb{E}_q R((s-\nu)/\alpha) = 1.$$

At this root, integrated Fenchel equality proves (15) and scalar dual attainment. Let $D_{f,\Psi}(z, \nu)$ denote the objective in (9). For any feasible w and finite dual value, expansion gives

$$\begin{aligned} D_{f,\Psi}(z,\nu) - \mathcal{G}(wq) = \Psi(m_w) + \Psi^*(z) - z^\top m_w \\ + \alpha \mathbb{E}_q[f(w) + f_+^*(u) - uw], \\ m_w = \mathbb{E}_q[w\phi], \qquad u = (r - z^\top \phi - \nu)/\alpha. \end{aligned} \tag{18}$$

Thus the two optimality conditions are the normalized conjugate response and $z \in \partial\Psi(m_w)$. For a proper closed convex extended-valued Ψ, it suffices that it be continuous at an attainable finite-penalty moment, relative to the moment affine space. Coverage entropy qualifies after removing feature coordinates that vanish q-almost surely. These sufficient conditions cover KL and half-Pearson; unbounded rewards and population reverse KL need separate integrability and attainment conditions. The finite-bank results do not require bounded population rewards.

Table 5: **Utility functionals and their FTFC calibration.** The first five rows use the Moment Dual (9); the last two calibrate a scalar tail threshold.

Utility $\mathcal{F}(p)$	Representation	Reward / calibration
Expected reward $\mathbb{E}_p r$	$\Psi = 0$; no features	$g^\star = r$
Moment matching $\mathbb{E}_p r - \frac{\gamma}{2}\|m_p - m_0\|^2$	ϕ: chosen descriptors $\Psi(m) = \frac{\gamma}{2}\|m - m_0\|^2$	$z^\star = \gamma(m^\star - m_0)$ $g^\star = r - z^{\star\top}\phi$
Coverage entropy $\mathbb{E}_p r + \gamma H(m_p)$	ϕ: region memberships $\Psi(m) = \gamma \sum_j m_j \log m_j$	$z_j^\star = \gamma(1 + \log m_j^\star)$ $g^\star = r - z^{\star\top}\phi$
D-optimal design $\log\det M_p, \quad r = 0$	$\phi(x) = v(x)v(x)^\top$ $\Psi(m) = -\log\det(M_0 + m)$	$g^\star = v^\top M_{p^\star}^{-1} v$
Constraint barrier† $\mathbb{E}_p r + \gamma \log(B - \mathbb{E}_p c)$	$\phi(x) = c(x)$ $\Psi(m) = -\gamma \log(B - m)$	$z^\star = \gamma/(B - m^\star)$ $g^\star = r - z^\star c$
Lower-tail CVaR $L_\tau(p) = \sup_\eta\{\eta + \mathbb{E}_p g_\eta^-\}$	$g_\eta^-(x) = -(\eta - r(x))_+/\tau$	Maximize $\eta + T_f(g_\eta^-)$ over observed reward knots.
Upper-tail CVaR $U_\tau(p) = \inf_\eta\{\eta + \mathbb{E}_p g_\eta^+\}$	$g_\eta^+(x) = (r(x) - \eta)_+/\tau$	Minimize $\eta + T_f(g_\eta^+)$ over the scalar threshold.

$m_p = \mathbb{E}_p\phi$, $m^\star = m_{p^\star}$, $\gamma > 0$, $H(m) = -\sum_j m_j \log m_j$, and $M_p = M_0 + \mathbb{E}_p[vv^\top]$ with $M_0 \succ 0$. The tail mass is $\tau \in (0, 1)$; $T_f(s) = \sup_{p \ll q}\{\mathbb{E}_p s - \alpha D_f(p\|q)\}$ is the optimized fixed-reward value. †The barrier uses the maximization convention, with $\mathbb{E}_p c < B$. Coverage entropy concerns fixed memberships, not the full continuous density.

Examples of fixed features. Expected reward has $\Psi = 0$ and $g = r$, so no feature vector is needed. For nonlinear utilities, examples with $\gamma > 0$ are:

Utility	Features $\phi(x)$	Penalty $\Psi(m)$
Coverage entropy	Region memberships	$\gamma \sum_j m_j \log m_j$
Moment matching	Chosen descriptors	$\frac{\gamma}{2}\|m - m_0\|^2$
D-optimal design ($r = 0$)	$v(x)v(x)^\top$	$-\log\det(M_0 + m)$

For coverage, $\phi_j \geq 0$ and $\sum_j \phi_j = 1$, so m_j is a region's aggregate mass. The rings use 16 soft radial-basis memberships (Appendix C.1.2); marginal costs discourage overrepresented regions. This is region entropy, not differential entropy of the full density. Moment matching instead specifies desired descriptor means m_0. The conjugates are given below. For experimental design, $v(x)$ is a fixed experiment feature vector and $M_p = M_0 + \mathbb{E}_p[vv^\top]$, $M_0 \succ 0$. The utility is $\log\det M_p$, with $g_p(x) = v(x)^\top M_p^{-1} v(x)$; matrix inner products replace $z^\top\phi$ (Boyd & Vandenberghe, 2004, Section 7.5).

For $\gamma > 0$ and a desired moment m_0, the structured dual uses

$$\Psi(m) = \tfrac{\gamma}{2}\|m - m_0\|^2 \qquad \Longrightarrow \qquad \Psi^*(z) = z^\top m_0 + \tfrac{1}{2\gamma}\|z\|^2,$$

$$\Psi(m) = \gamma \sum_j m_j \log m_j, \quad m \in \Delta_k \qquad \Longrightarrow \qquad \Psi^*(z) = \gamma \log \sum_j e^{z_j/\gamma}.$$

At an optimum, $z^\star \in \partial\Psi(m_{p^\star})$. For the entropy penalty, marginal costs are defined up to a common additive constant on the simplex. Under KL, eliminating ν gives

$$D_{\mathrm{KL},\Psi}(z) = \Psi^*(z) + \alpha \log \mathbb{E}_q e^{(r - z^\top\phi)/\alpha}, \qquad \nabla D_{\mathrm{KL},\Psi}(z) = \nabla\Psi^*(z) - \mathbb{E}_{w_{g_z} q}\phi.$$

At an optimum, the moment implied by the conjugate equals the weighted feature mean.

Utility examples. Table 5 writes the utility families in FTFC notation, using examples from De Santi et al. (2025, Table 1). It distinguishes the convex Moment Dual (9), scalar outer optimization, and objectives with full-density dependence. The statements concern endpoint calibration; native fitting and bank generalization remain separate.

Entropy and coverage. For a finite outcome space, $\phi_j(x) = \mathbf{1}\{x = x_j\}$ gives $m_j = p(x_j)$, so $\Psi(m) = \gamma \sum_j m_j \log m_j$ represents $\gamma H(p)$ exactly with $r = 0$. Region indicators instead give the entropy of region masses. In a continuous space, $H(p) = -\int p(x) \log p(x)\, d\mu(x)$ depends on the entire density relative to the reference measure μ; its marginal reward is $-1 - \log p(x)$ where

defined. The functional dual can apply under its regularity assumptions, but fixed region features do not exactly represent this differential entropy or supply its density values.

Experimental design. For any concave matrix criterion s, choose $\phi(x) = v(x)v(x)^\top$ and $\Psi(m) = -s(M_0 + m)$, with matrix inner products in place of dot products. Then $\mathcal{F}(p) = s(M_p)$ and $g_p(x) = \langle \nabla s(M_p), v(x)v(x)^\top \rangle$ when differentiable. The choices $s(M) = \log\det M$ and $s(M) = -\operatorname{tr}(M^{-1})$ yield $g_p(x) = v(x)^\top M_p^{-1} v(x)$ and $g_p(x) = v(x)^\top M_p^{-2} v(x)$, respectively. The criterion $s(M) = -\lambda_{\max}(M)$ is also concave and uses a supergradient at repeated eigenvalues. We use $M_0 \succ 0$ to ensure the domain of the log-determinant and inverse criteria; other fixed offsets require checking that domain.

Constraint penalties and barrier signs. Any convex penalty of the mean cost, $\Psi(\mathbb{E}_p c)$, fits the moment class. For the upper bound $\mathbb{E}_p c < B$, the maximization barrier is $\mathcal{F}(p) = \mathbb{E}_p r + \gamma \log(B - \mathbb{E}_p c)$: $\Psi(m) = -\gamma \log(B - m)$ is convex, and the marginal cost $z = \gamma/(B - m)$ grows as the budget is approached. This convention differs from the expression $\mathbb{E}_p r - \gamma \log(\mathbb{E}_p c - B)$ printed in FDC's table. The latter would require $\Psi(m) = \gamma \log(m - B)$, which is concave, so it is not an instance of the convex Moment Dual.

Mean–variance reward. For $\mathcal{F}(p) = \mathbb{E}_p r - \gamma \operatorname{Var}_p(r)$, $\phi = (r, r^2)$ gives $\Psi(m) = \gamma(m_2 - m_1^2)$. This representation is exact, but Ψ is not convex. The marginal reward is $g_p = r - \gamma r^2 + 2\gamma(\mathbb{E}_p r)r$; Equation (9) does not provide a convex calibration for it. There is instead a scalar outer reduction:

$$\mathcal{F}(p) = \sup_a \mathbb{E}_p[r - \gamma(r - a)^2],$$
$$\sup_p \{\mathcal{F}(p) - \alpha D_f(p\|q)\} = \sup_a T_f(r - \gamma(r - a)^2).$$

Indeed, $\mathbb{E}_p(r-a)^2 = \operatorname{Var}_p(r) + (\mathbb{E}_p r - a)^2$; the maximizing center is $a = \mathbb{E}_p r$, and the two suprema commute. On a finite bank, a can be restricted to the observed reward range. The outer objective need not be concave, and the CVaR reward-knot rule does not apply.

Conditional mode objectives. Let fixed condition masses be $\rho_h > 0$ and $\bar{p} = \sum_h \rho_h p_h$. The negative-sign expression in FDC's table is $-\sum_h \rho_h \mathrm{D}_{\mathrm{KL}}(p_h\|\bar{p})$. On finite outcome cells A_j, use the joint law $P(h, x) = \rho_h p_h(x)$ and features $\phi_{hj}(h', x) = \mathbf{1}\{h' = h,\, x \in A_j\}$. Writing $m_{hj} = P(h, A_j)$ and $\bar{m}_j = \sum_h m_{hj}$, its discrete version has $r = 0$ and the convex penalty

$$\Psi(m) = \sum_{h,j} m_{hj} \log \frac{m_{hj}}{\rho_h \bar{m}_j}, \qquad g_P(h, x) = -\log \frac{m_{hj}}{\rho_h \bar{m}_j} \quad (x \in A_j).$$

Convexity follows from joint convexity of relative entropy with fixed ρ_h; normalization uses one multiplier per condition. For continuous conditionals, the exact functional retains their full density ratios. The sign matters: maximizing the negative expression reduces conditional separation, whereas maximizing positive mutual information encourages distinct modes and reverses the convex penalty's sign. The latter is not covered by the convex Moment Dual.

Tail objectives. The lower- and upper-tail rows use the exact threshold representations in (23), including fractional mass at atoms. They require no moment-feature vector. Their outer optimization directions differ: lower CVaR maximizes its threshold objective and upper CVaR minimizes it, as derived in (24). Proposition 3 supplies the exact finite-bank search for the lower tail.

A.2 Finite-bank optimization

Let $a_i > 0$ be reference masses on cached endpoints x_i, $\sum_i a_i = 1$. For condition groups I_h, keep their reference probabilities $\rho_h = \sum_{i \in I_h} a_i$ fixed. Target masses b_i and ratios w_i satisfy

$$\mathcal{C} = \{b \geq 0 : \sum_{i \in I_h} b_i = \rho_h \text{ for every } h\}, \qquad b_i = a_i w_i,$$
$$\mathcal{G}_N(b) = \mathcal{F}_N(b) - \alpha \sum_i a_i f(b_i/a_i). \tag{19}$$

Without conditioning, there is one group, so $\mathcal{C}$ is the simplex. The group multipliers enter the Lagrangian as $\sum_h \nu_h(\rho_h - \sum_{i \in I_h} b_i)$. Maximizing each ratio $b_i/a_i \geq 0$ independently yields

$$D_{f,N}(g, \nu) = C_N(g) + \sum_h \rho_h \nu_h + \alpha \sum_i a_i f_+^*((g_i - \nu_{h_i})/\alpha),$$
$$C_N(g) = \sup_{b \in \mathcal{C}} \{\mathcal{F}_N(b) - b^\top g\}, \qquad b_i \in a_i \partial f_+^*((g_i - \nu_{h_i})/\alpha). \tag{20}$$

Choose ν so $b \in \mathcal{C}$. Theorem 1 applies with sums in place of expectations. For $\mathcal{F}_N(b) = r^\top b - \Psi(\sum_i b_i\phi_i)$, substitute $g_i = r_i - z^\top\phi_i$ and the envelope offset $\Psi^*(z)$. For proper closed convex Ψ, the finite-dimensional qualification is $b^\circ \in \mathrm{ri}\,\mathcal{C}$ with $\sum_i b_i^\circ\phi_i \in \mathrm{ri}\,\mathrm{dom}\,\Psi$; the preceding Fenchel argument then applies.

Solving for z and ν. For an unconditional bank, cache $x_i \sim q$, rewards r_i, and features $\phi_i = \phi(x_i)$, with reference masses $a_i = 1/N$. Replace $\mathbb{E}_q$ in the middle line of (9) by $\sum_i a_i$ to obtain $D_N(z,\nu)$. For each z, choose ν to normalize the conjugate response:

$$w_i(z) \in \partial f_+^*((r_i - z^\top\phi_i - \nu)/\alpha), \qquad \sum_i a_i w_i(z) = 1.$$

Under the response assumptions in Appendix A.1, this is a monotone scalar equation; for KL, log-sum-exp gives the solution. Minimize the resulting convex objective over z, whose gradient, when defined, is

$$\nabla_z\Big[\min_\nu D_N(z,\nu)\Big] = \nabla\Psi^*(z) - \sum_i a_i w_i(z)\phi_i.$$

Calibration matches the statistics implied by z to those induced by its weights. It uses cached endpoints, without updating the generator. With fixed groups, use one normalizer per group as in (20); the moment gradient is unchanged.

For KL, $Z_{z,h} = \sum_{i\in I_h}(a_i/\rho_h)e^{(r_i - z^\top\phi_i)/\alpha}$ eliminates the normalizers:

$$\begin{aligned} D_N(z) &= \Psi^*(z) + \alpha\sum_h \rho_h \log Z_{z,h}, \\ b_{z,i} &= a_i e^{(r_i - z^\top\phi_i)/\alpha}/Z_{z,h_i}, \qquad m_b = \sum_i b_i\phi_i, \\ D_N(z) - \mathcal{G}_N(b) &= \alpha \mathrm{D}_{\mathrm{KL}}(b\|b_z) + \Psi(m_b) + \Psi^*(z) - z^\top m_b. \end{aligned} \tag{21}$$

The gradient, when defined, is $\nabla\Psi^*(z) - m_{b_z}$. The moment and entropy conjugates are in Appendix A.1. A global utility shares its z or tail threshold across groups; separate conditional utilities are different objectives.

Tail rewards in the image and molecule benchmarks. CVaR uses an exact scalar-threshold representation instead of fixed moment features. For tail mass τ, its lower- and upper-tail pseudo-rewards are

$$g_c^{\mathrm{lower}}(x) = -\frac{(c - r(x))_+}{\tau}, \qquad g_c^{\mathrm{upper}}(x) = \frac{(r(x) - c)_+}{\tau}. \tag{22}$$

The threshold c is calibrated using (24): upper-CVaR minimizes its threshold objective; lower-CVaR maximizes it and is not covered by the concave utility theorem. On a finite bank, the latter maximum is attained at an observed reward (Appendix A.2). These objectives recover TFFT under KL (Wang et al., 2026). In Stable Diffusion, calibration uses $r = 100\,\mathrm{ImageReward}$, $\alpha = 1$, and lower-tail mass 0.2. In QM9, it uses $r = -E_{\mathrm{GFN1}}/100$, $\alpha = 0.01$, and upper-tail mass 0.002. Both optimize a scalar threshold rather than moment features. Prompt and atom-count probabilities remain fixed, respectively: for group h, one normalizer enforces $\sum_{i\in I_h} a_i w_i = \rho_h$, while the threshold is shared across groups.

CVaR calibration. For tail mass $\tau \in (0,1)$, the lower and upper reward averages are

$$L_\tau(p) = \sup_c\{c - \tau^{-1}\mathbb{E}_p(c-r)_+\}, \qquad U_\tau(p) = \inf_c\{c + \tau^{-1}\mathbb{E}_p(r-c)_+\}. \tag{23}$$

These definitions count fractional mass at threshold atoms. Their threshold-dependent rewards are $-(c-r)_+/\tau$ and $(r-c)_+/\tau$. Upper-CVaR is concave in the law; lower-CVaR is convex. Using the fixed-reward response (15) gives

$$\begin{aligned} V_U &= \min_c\{c + T_f((r-c)_+/\tau)\}, \\ V_L &= \max_c\{c + T_f(-(c-r)_+/\tau)\}. \end{aligned} \tag{24}$$

For the upper tail, restrict c to the reward range and apply convex–concave minimax under the finite-bank assumptions. For the lower tail, commute two suprema. With KL, (16) recovers TFFT's threshold objectives (Wang et al., 2026). The lower-tail objective need not be concave; its exact finite-bank solution follows instead from piecewise convexity.

Proposition 3 (Lower-CVaR requires only reward knots)**.** *Let $\mathcal{B}$ be a nonempty compact subset of the probability simplex, R lower semicontinuous, bounded below and finite somewhere on $\mathcal{B}$, and r_i finite. For $0 < \tau < 1$ and $T_R(s) = \sup_{b\in\mathcal{B}}\{b^\top s - R(b)\}$,*

$$\max_{b\in\mathcal{B}}\{L_\tau(b) - R(b)\} = \max_{c\in\{r_1,\dots,r_N\}}\{c + T_R(-(c\mathbf{1} - r)_+/\tau)\}. \tag{25}$$

An exact response at a maximizing knot attains the left side.

Proof. Commuting the two suprema gives $H(c) = c + T_R(-(c\mathbf{1} - r)_+/\tau)$. The response value T_R is convex and $T_R(s + d\mathbf{1}) = T_R(s) + d$. Hence

$$H(c) = \begin{cases} c + T_R(0), & c \le \min_i r_i, \\ \text{convex on each } [r_{(j)}, r_{(j+1)}], & \min_i r_i \le c \le \max_i r_i, \\ (1 - 1/\tau)c + T_R(r/\tau), & c \ge \max_i r_i. \end{cases}$$

The middle claim follows because each hinge is affine between knots. The outer pieces increase and decrease, respectively, so a knot maximizes H. Compactness and lower semicontinuity give an attained response there; it maximizes the joint objective over (b, c). □

For unconditional KL and sorted rewards $r_{(1)} \le \cdots \le r_{(N)}$, each knot normalizer is

$$Z(c) = e^{-c/(\alpha\tau)} \sum_{r_i \le c} a_i e^{r_i/(\alpha\tau)} + \sum_{r_i > c} a_i, \qquad H(c) = c + \alpha \log Z(c).$$

Sorting plus log-domain prefix/suffix sums costs $O(N \log N)$. For general f, the same knot search is exact but requires its own normalization solves. At $\alpha = 0$, solve the unregularized mass problem; the scaled conjugate formulas do not apply.

Optional score optimization. If no compact dual is available, a neural score model can optimize the full-bank objective through normalized scores. For positive masses, $g_i = \partial\mathcal{F}_N/\partial b_i$ and $u_i = g_i - \alpha f'(b_i/a_i)$, differentiation gives

$$\begin{aligned} b_i &= \frac{a_i e^{s_\psi(x_i)}}{\sum_{j\in I_{h_i}} (a_j/\rho_{h_i}) e^{s_\psi(x_j)}}, \\ \nabla_\psi \mathcal{G}_N &= \sum_i b_i (u_i - \bar{u}_{h_i}) \nabla_\psi s_\psi(x_i), \qquad \bar{u}_h = \rho_h^{-1} \sum_{i\in I_h} b_i u_i. \end{aligned} \tag{26}$$

Indeed, within a group, $\partial b_i/\partial s_j = b_i(\mathbf{1}_{i=j} - b_j/\rho_h)$; different groups have zero cross-derivatives. Chunking this sum with fixed full-bank coefficients preserves the gradient. For a concave utility and a supergradient g at b, the envelope is tight there. Theorem 1 therefore supplies a bound on target suboptimality; under KL it is $\alpha \mathrm{D}_{\mathrm{KL}}(b\|b_g)$. Here $b_{g,i} = a_i e^{g_i/\alpha} / \sum_{j\in I_{h_i}} (a_j/\rho_{h_i}) e^{g_j/\alpha}$. At $\alpha = 0$ the bound is $\sum_h \rho_h \max_{i\in I_h} g_i - g^\top b$. For nonconcave utilities these tangent expressions do not imply global optimality.

Finite-bank approximation. The gap bounds optimization error for the discrete reference $q_N = \sum_i a_i \delta_{x_i}$. It does not bound population or generator error. Conditioning a bank on reward eligibility replaces q_h by $q_h(\cdot \mid \text{eligible})$; changing deployment condition frequencies changes the joint law. Optimization accuracy alone therefore does not establish sample coverage or agreement with the population target. The eligibility rules and conditioning groups for each benchmark are given in Appendix B.

A.3 From endpoint weights to a generative model

Corollary 4 (Optimality under exact target realization)**.** *Let wq be ε-optimal for the endpoint relaxation. Assume finite second moments of the endpoint and source, and an admissible exact target predictor, obtained by native regression or diffusion guidance as above. For flows, assume the ODE and continuity equation have matching unique marginals. For diffusion, assume a well-posed reverse process, initialized at the target's correct noisy law and integrated exactly. Then*

$$\boxed{p_1^{\widehat{\pi}} = wq, \qquad 0 \le \sup_{\pi\in\Pi} \mathcal{G}(p_1^\pi) - \mathcal{G}(p_1^{\widehat{\pi}}) \le \varepsilon.} \tag{27}$$

Conditional regression identifies the target dynamics, and uniqueness identifies their endpoint law. The policy optimum is bounded by the endpoint relaxation; finite-bank estimation, imperfect fitting, and numerical sampling remain separate errors.

Proof of Theorem 2. Fix $p = wq$ and $Y_t \sim K_t(\cdot \mid X)$. Write $\widetilde{q}_t = K_t\#q$, $\widetilde{p}_t = K_t\#p$, and $h_t(y) = \mathbb{E}_q[w(X) \mid Y_t = y]$. Assume finite conditional moments and $0 < h_t < \infty$ on the relevant support. For any test function φ making the following expectations finite, conditioning under the reference joint law gives

$$\begin{aligned}\mathbb{E}_p\varphi(Y_t) &= \mathbb{E}_q[w(X)\varphi(Y_t)] = \mathbb{E}_{\widetilde{q}_t}[h_t(Y_t)\varphi(Y_t)],\\ \mathbb{E}_p[A_t\varphi(Y_t)] &= \mathbb{E}_{\widetilde{q}_t}[\mathbb{E}_q[w(X)A_t \mid Y_t]\varphi(Y_t)].\end{aligned}$$

Dividing the second conditional density by the first proves (12); logarithmic differentiation of $\widetilde{p}_t = h_t\widetilde{q}_t$ proves (13). For the Gaussian kernel,

$$\nabla_y \log \widetilde{p}_t(y) = \mathbb{E}_p\left[-\frac{y - a_t X}{\sigma_t^2}\,\middle|\, Y_t = y\right] = -\frac{\mathbb{E}_p[\epsilon \mid Y_t = y]}{\sigma_t},$$

and likewise under q, proving the noise-predictor correction. For $d(y,t) = \ell(y,t) - \log h_t(y)$, conditioning the in-sample loss gives

$$\begin{aligned}\mathcal{L}_{\mathrm{IGL}}(\ell; w) &= \mathbb{E}_{t,\widetilde{q}_t}[h_t e^{-\ell} + \ell],\\ \mathcal{L}_{\mathrm{IGL}}(\ell; w) - \mathcal{L}_{\mathrm{IGL}}(\log h; w) &= \mathbb{E}_{t,\widetilde{q}_t}[e^{-d} - 1 + d] \geq 0.\end{aligned}$$

Since $e^{-d} - 1 + d = 0$ exactly when $d = 0$, the unique minimizer is $\ell^\star = \log h$ almost everywhere. Its smooth representative has the score gradient in (14). Taking $w = w^\star$ proves the claim. This is the function-space argument of Mao et al. (2024, Theorem 1); neither neural optimization convergence nor score-gradient accuracy follows from a small loss alone. □

Proof of Corollary 4. For $I_t = (1-t)\epsilon + tX$, $X \sim p = wq$ independent of $\epsilon \sim p_0$, let $\mu_t = \mathrm{Law}(I_t)$ and $v_p(y,t) = \mathbb{E}_p[X - \epsilon \mid I_t = y]$. Change of measure and regression orthogonality give

$$\begin{aligned}\mathcal{R}(\pi; w) &= \mathbb{E}_{t,X\sim p,\epsilon\sim p_0}\|\pi(I_t,t) - (X-\epsilon)\|^2,\\ \mathcal{R}(\pi; w) - \mathcal{R}(v_p; w) &= \int_0^1 \mathbb{E}_{\mu_t}\|\pi(Y_t,t) - v_p(Y_t,t)\|^2\,dt.\end{aligned}$$

The cross term vanishes because $\mathbb{E}_p[X - \epsilon - v_p(I_t,t) \mid I_t] = 0$. Since $v_p \in \Pi$, a global minimizer $\widehat{\pi}$ equals v_p under $dt\,\mu_t$ almost everywhere. For every smooth compactly supported φ,

$$\frac{d}{dt}\mathbb{E}_p\varphi(I_t) = \mathbb{E}_p[\nabla\varphi(I_t)^\top(X-\epsilon)] = \int \nabla\varphi(y)^\top v_p(y,t)\,\mu_t(dy).$$

Thus μ_t solves the continuity equation for both v_p and $\widehat{\pi}$, with $\mu_0 = p_0$ and $\mu_1 = p$. The assumed uniqueness and consistency with ODE transport imply $p_1^{\widehat{\pi}} = p$. Since every admissible policy is feasible for the endpoint relaxation and wq is ε-optimal for that relaxation,

$$0 \leq \sup_{\pi\in\Pi}\mathcal{G}(p_1^\pi) - \mathcal{G}(p_1^{\widehat{\pi}}) \leq \varepsilon.$$

For diffusion, regression or Theorem 2 yields the exact target score. A well-posed reverse process, initialized at the target's correct noisy law and integrated exactly, then has endpoint p and the same objective bound. If the terminal noising coefficient $a_T = 0$, this initial noisy law is the common Gaussian $\mathcal{N}(0, \sigma_T^2 I)$. Otherwise, replacing it by a Gaussian introduces an additional approximation. The guidance route also requires the exact reference score of the same noised q used for calibration. □

Fitting on a finite bank. For frozen bank masses, the training objective and its two equivalent sampling forms are, with t and ϵ sampled as in (11),

$$\begin{aligned}\mathcal{R}_N(\theta; b) &= \sum_i b_i \mathbb{E}_{t,\epsilon}[\ell_\theta(x_i,t,\epsilon)]\\ &= \mathbb{E}_{i\sim b,\,t,\epsilon}[\ell_\theta(x_i,t,\epsilon)]\\ &= \mathbb{E}_{i\sim a,\,t,\epsilon}[(b_i/a_i)\ell_\theta(x_i,t,\epsilon)].\end{aligned} \tag{28}$$

Fresh native noise preserves the chosen source law. If Q is the original generation-path law with source p_0, weighting its paths by $w(X_1)$ instead gives source marginal $p_0(dx_0)\mathbb{E}_Q[w(X_1) \mid X_0 = x_0]$, generally different from p_0. If native preprocessing uses a noninvertible map T, fitting targets $T_\#(wq)$; any exact utility guarantee must use that same observable. The corollary concerns exact target predictors and exact sampling. Finite-bank estimation, restricted adapters, and time discretization remain separate errors.

Guidance for Gaussian flow interpolations. For $Y_t = a_t X + b_t \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$ independent of X, write $\bar{x}_q(y) = \mathbb{E}_q[X \mid Y_t = y]$ and $\bar{x}_p(y) = \mathbb{E}_p[X \mid Y_t = y]$. For $a_t, b_t > 0$, Gaussian differentiation gives

$$\begin{aligned}
\nabla \log \widetilde{q}_t(y) &= (a_t \bar{x}_q(y) - y)/b_t^2, \\
\nabla \log h_t(y) &= a_t(\bar{x}_p(y) - \bar{x}_q(y))/b_t^2, \\
\epsilon_p - \epsilon_q &= -a_t(\bar{x}_p - \bar{x}_q)/b_t = -b_t \nabla \log h_t, \\
v_p - v_q &= (\dot{a}_t - a_t \dot{b}_t/b_t)(\bar{x}_p - \bar{x}_q) = \left(\frac{\dot{a}_t b_t^2}{a_t} - b_t \dot{b}_t \right) \nabla \log h_t.
\end{aligned}$$

Here b_t plays the role of the diffusion noise scale σ_t. For a Gaussian flow interpolation, the correction is

$$v_p = v_q + \kappa_t \nabla \log h_t, \qquad \kappa_t = \frac{\dot{a}_t b_t^2}{a_t} - b_t \dot{b}_t. \tag{29}$$

For $(a_t, b_t) = (t, 1-t)$, $\kappa_t = (1-t)/t$ on $0 < t < 1$; endpoint values require suitable limits. These identities assume finite conditional moments and differentiation under the integral. The reference predictor must correspond to this same corruption or interpolation; arbitrary pretrained velocity fields cannot be corrected by this formula without that compatibility (Feng et al., 2025; Zhang et al., 2025).

A.4 Reference-preserving regression by changing the training label

Scope. We analyze an optional fitting-loss modification that leaves Fenchel calibration unchanged. It accounts for the gap between the pretrained predictor and the conditional mean of generated endpoints under, e.g., classifier-free guidance. The modification is not used in the benchmarks; the derivation alone does not imply better generation quality.

Retain the native pair (Y_t, A_t) from (11), a frozen normalized ratio $w = dp/dq$, and a frozen predictor $m_0(y, t)$. Write $\delta_\theta = m_\theta - m_0$ and let $\lambda > 0$. All expectations below use the same fresh corruption or interpolation conditional on the endpoint, and have finite displayed second moments. Consider the centered quadratic risk

$$\mathcal{L}_\lambda(\theta) = \mathbb{E}_{q,t,\epsilon}\left[(w + \lambda)\|\delta_\theta\|^2 + 2(w-1)\langle \delta_\theta, m_0 - A_t \rangle\right]. \tag{30}$$

The quadratic coefficient is positive even where $w = 0$. If $w \equiv 1$ and $m_\theta = m_0$ pointwise, each example has zero parameter gradient; ordinary regression to fresh native labels need not have this property for an imperfect teacher.

Proposition 5 (An unweighted regression form)**.** *Define the endpoint mixture and corrected native label*

$$\begin{aligned}
s_\lambda &= \frac{p + \lambda q}{1 + \lambda}, & a_\lambda(X) &= \frac{w(X) - 1}{w(X) + \lambda}, \\
\widetilde{A}_t &= m_0(Y_t, t) + a_\lambda(X)\big(A_t - m_0(Y_t, t)\big).
\end{aligned} \tag{31}$$

Then the ordinary squared regression loss

$$\mathcal{J}_\lambda(\theta) := \mathbb{E}_{X \sim s_\lambda, t, \epsilon} \|m_\theta(Y_t, t) - \widetilde{A}_t\|^2 = \frac{\mathcal{L}_\lambda(\theta)}{1 + \lambda} + C_\lambda \tag{32}$$

has a finite constant C_λ independent of θ. No additional endpoint weight multiplies this loss. Let $h_t(y) = \mathbb{E}_q[w(X) \mid Y_t = y]$ and $m_q(y, t) = \mathbb{E}_q[A_t \mid Y_t = y]$. Its unrestricted conditional optimum is

$$\begin{aligned}
m_\lambda^*(y, t) &= \frac{h_t m_p + (1 + \lambda) m_0 - m_q}{h_t + \lambda}, \\
m_\lambda^* - m_{s_\lambda} &= \frac{1 + \lambda}{h_t + \lambda}(m_0 - m_q), \qquad m_{s_\lambda} = \frac{h_t m_p + \lambda m_q}{h_t + \lambda}.
\end{aligned} \tag{33}$$

Here $h_t m_p$ means $\mathbb{E}_q[w(X) A_t \mid Y_t = y]$, also when $h_t = 0$. In particular, an exact reference teacher recovers the native predictor of s_λ, rather than of the unmixed target p.

Proof. Expanding the square in (32) and using $ds_\lambda/dq = (w+\lambda)/(1+\lambda)$ gives

$$C_\lambda = \frac{1}{1+\lambda}\mathbb{E}_q\left[\frac{(w-1)^2}{w+\lambda}\|A_t - m_0\|^2\right].$$

The remaining terms are exactly $\mathcal{L}_\lambda/(1+\lambda)$. The teacher and calibrated ratios are held fixed when differentiating. Conditional regression gives $m^*_\lambda = \mathbb{E}_{s_\lambda}[\widetilde{A}_t \mid Y_t]$. Since the conditional normalizer is $h_t + \lambda$,

$$\mathbb{E}_{s_\lambda}[a_\lambda(A_t - m_0) \mid Y_t] = \frac{h_t m_p - m_q - (h_t - 1)m_0}{h_t + \lambda}.$$

Adding m_0 proves the first identity; subtracting the mixture's conditional mean proves the second. □

Symmetric mixture. At $\lambda = 1$, draw an endpoint from p or q with equal probability and use

$$a_1(X) = \frac{w(X) - 1}{w(X) + 1} = \tanh\left(\tfrac{1}{2}\log w(X)\right) \in [-1, 1]. \tag{34}$$

For $w = 0$, the equality uses the limit $a_1 = -1$. Only a bounded scalar modifies the native label; the ratio network is absent at inference. In general $-1/\lambda \le a_\lambda \le 1$. The mixture strength is a parameter of the endpoint law being fitted, not just of its estimator. Shared conditioning-group masses are preserved, and any positive time weight may multiply both objectives without changing the argument.

Posterior averaging removes component-label noise. One way to sample s_λ is to draw a component variable $B \in \{1, -1/\lambda\}$ with probabilities $1/(1+\lambda)$ and $\lambda/(1+\lambda)$, then draw X from p or q, respectively. Bayes' rule gives

$$\Pr(B = 1 \mid X) = \frac{w(X)}{w(X) + \lambda}, \qquad \mathbb{E}[B \mid X] = a_\lambda(X).$$

The random label $m_0 + B(A_t - m_0)$ has conditional expectation $\widetilde{A}_t$ given the endpoint and its fresh native randomness. For a fixed student and teacher, the squared-loss parameter gradient is affine in this label. Consequently its soft-label gradient is the conditional expectation of the random-component gradient. When these gradients have finite second moments, the law of total covariance gives

$$\mathrm{Cov}(G_{\mathrm{soft}}) \preceq \mathrm{Cov}(G_{\mathrm{component}}). \tag{35}$$

This Rao–Blackwell bound compares the two label estimators under the same mixture. It does not compare their variance with ordinary target resampling, reference sampling, or other importance proposals, and it does not imply identical Adam updates.

Error bound. Let $\widetilde{s}_{\lambda,t}$ and $\widetilde{q}_t$ be the corresponding noisy or interpolated laws. Integrating (33) and using $d\widetilde{s}_{\lambda,t}/d\widetilde{q}_t = (h_t + \lambda)/(1+\lambda)$ yields

$$\mathbb{E}_{t,\widetilde{s}_{\lambda,t}}\|m^*_\lambda - m_{s_\lambda}\|^2 \le \frac{1+\lambda}{\lambda}\mathbb{E}_{t,\widetilde{q}_t}\|m_0 - m_q\|^2. \tag{36}$$

For $\Delta_\theta = \mathcal{J}_\lambda(\theta) - \inf_m \mathcal{J}_\lambda(m)$, the infimum over all square-integrable predictors, regression orthogonality and the triangle inequality give

$$\|m_\theta - m_{s_\lambda}\|_{L^2(dt\,\widetilde{s}_{\lambda,t})} \le \sqrt{\Delta_\theta} + \sqrt{\frac{1+\lambda}{\lambda}}\,\|m_0 - m_q\|_{L^2(dt\,\widetilde{q}_t)}. \tag{37}$$

For the symmetric choice the squared-error amplification bound is two. The bound does not estimate the actual teacher error. Even with an exact teacher and fit, the endpoint guarantee requires the assumptions of Corollary 4 for s_λ; this construction is neither an exact realization of p nor a guarantee of improving the paper's utility.

A.5 A CENTERED REGRESSION ON THE REFERENCE SAMPLER'S TRANSITIONS

Scope. This optional realization uses saved transitions from the reference sampler instead of fresh endpoint corruption. Fenchel calibration is unchanged, and the approximate-teacher term of Appendix A.4 vanishes at the population level. It is not used in the benchmark tables; the identities alone do not imply better generation quality.

Utility and divergence are inputs to calibration. The endpoint target still solves $\mathcal{F}(p)-\alpha D_f(p\|q)$ in (3), under the conditions of Section 4. The construction below takes only its normalized nonnegative ratio w as input; it does not assume expected reward or KL weights, so half-Pearson responses with zeros are allowed. The path KL is the Gaussian transition-fitting discrepancy, distinct from the endpoint f-divergence. This backend is optional; the flow and denoising realizations in Section 4.5 remain available.

Fix a conditioning value and suppress it in the notation. Let Q be a Markov reference path law with transitions

$$X_{t+1} = \mu_{q,t}(X_t) + \sigma_t \xi_t, \qquad \xi_t \sim \mathcal{N}(0, I), \quad \sigma_t > 0, \qquad t = 0, \dots, T-1.$$

Each innovation is independent of the preceding path. Write $P = w(X_T)Q$, where $w \geq 0$ and $\mathbb{E}_Q w = 1$, and $S_\lambda = (P + \lambda Q)/(1+\lambda)$ for $\lambda > 0$. The model Q_θ retains Q's initial law and transition covariance, and changes the transition mean to $\mu_{q,t}(X_t) + \sigma_t r_\theta(X_t, t)$. The correction must not observe its current innovation. Assume that the displayed risks and relative entropies are finite.

Proposition 6 (Centering and a conditional mixture correction)**.** *The centered risk*

$$\mathcal{L}_\lambda^{\mathrm{path}}(\theta) = \mathbb{E}_Q \sum_{t=0}^{T-1} \left[\tfrac{1}{2}(w+\lambda)\|r_\theta(X_t,t)\|^2 - (w-1)\langle \xi_t, r_\theta(X_t,t)\rangle \right] \tag{38}$$

satisfies

$$\mathcal{L}_\lambda^{\mathrm{path}}(\theta) = (1+\lambda)\left[\mathrm{D_{KL}}(S_\lambda \| Q_\theta) - \mathrm{D_{KL}}(S_\lambda \| Q)\right]. \tag{39}$$

Define $h_t(x) = \mathbb{E}_Q[w(X_T) \mid X_t = x]$ *and* $a_t(x) = \mathbb{E}_Q[w(X_T)\xi_t \mid X_t = x]$. *The unrestricted conditional minimizer is*

$$r_\lambda^*(x,t) = \frac{a_t(x)}{h_t(x)+\lambda} = \frac{h_t(x)}{h_t(x)+\lambda}\, r_P^*(x,t), \qquad r_P^*(x,t) = \mathbb{E}_P[\xi_t \mid X_t = x], \tag{40}$$

where the second equality applies when $h_t(x) > 0$. *When* $h_t(x) = 0$, $a_t(x) = 0$ *and* $r_\lambda^*(x,t) = 0$.

Proof. The Gaussian likelihood ratio gives $\log(dQ/dQ_\theta) = \sum_t [\|r_\theta\|^2/2 - \langle \xi_t, r_\theta\rangle]$. Since r_θ is adapted, $\mathbb{E}_Q\langle \xi_t, r_\theta(X_t,t)\rangle = 0$. Multiplying the likelihood ratio by $w+\lambda$ and adding this zero-mean term proves (39). Conditional on $X_t = x$, the part depending on r is $\frac{1}{2}(h_t+\lambda)\|r\|^2 - a_t^\top r$. Its positive curvature gives the unique minimizer. Bayes' rule gives $a_t = h_t \mathbb{E}_P[\xi_t \mid X_t]$ when $h_t > 0$; nonnegativity of w gives $a_t = 0$ when $h_t = 0$. □

A correction weighted by the current state. The factor $h_t/(h_t+\lambda)$ is the posterior probability of the target component in S_λ given X_t, so the ideal correction shrinks toward the reference where target mass is small. This describes the conditional optimum, not a bound on neural predictions or transition displacement. The regression fits it directly and needs no estimate of h_t at inference. For another interpretation, set $H_t(m) = \mathbb{E}_\xi[h_{t+1}(m + \sigma_t \xi)]$, so $h_t(x) = H_t(\mu_{q,t}(x))$ by the Markov property. When differentiation under this Gaussian integral is justified,

$$r_\lambda^*(x,t) = \sigma_t \nabla_m \log\big(H_t(m)+\lambda\big)\big|_{m=\mu_{q,t}(x)}. \tag{41}$$

The derivative is with respect to the transition mean m, not generally the state x; replacing it by a state gradient omits a Jacobian.

Sampling and deployment. Completing the square in (38) gives label $(w-1)\xi_t/(w+\lambda)$ and curvature $w+\lambda$. Sampling paths from $D = (P+Q)/2$ and multiplying both terms by $dQ/dD = 2/(1+w)$ preserves the risk. For $0 < \lambda \leq 1$, the corrected curvature lies in $[2\lambda, 2]$ and the signed linear coefficient in $[-2, 2]$. These bounds do not imply lower neural-gradient variance. For a Gaussian diffusion step with classifier-free guidance g, write

$$\mu_q = A_t X_t + C_t[(1-g)\epsilon_q^- + g\epsilon_q^+], \qquad r_\theta = \frac{gC_t}{\sigma_t}(\epsilon_\theta^+ - \epsilon_q^+).$$

Then the reference positive prediction cancels:

$$\mu_q + \sigma_t r_\theta = A_t X_t + C_t[(1-g)\epsilon_q^- + g\epsilon_\theta^+]. \tag{42}$$

Training uses a frozen reference positive prediction, but deployment needs only the frozen negative and learned positive branches. The branch count alone does not establish wall-time parity with ordinary guidance.

Calibrating the mixture for a general objective. Fitting S_λ after calibrating P changes the endpoint target. An alternative is to calibrate the desired mixture directly. Put $\varrho = \lambda/(1+\lambda)$ and solve

$$\sup_{v \geq \varrho,\, \mathbb{E}_q v = 1} \{\mathcal{F}(vq) - \alpha \mathbb{E}_q f(v)\}, \qquad w = \frac{v - \varrho}{1 - \varrho}. \tag{43}$$

Then $w \geq 0$, $\mathbb{E}_q w = 1$, and $(w+\lambda)/(1+\lambda) = v$. Supplying this auxiliary ratio w to (38) therefore projects toward the target calibrated for the specified $\mathcal{F}$ and f. This is the original endpoint objective with an explicit reference floor, not the unconstrained problem or a replacement of its divergence by KL. Fixed condition masses are preserved by the same affine transformation. For a utility admitting the supporting envelope $C(g)$ from (6), replace f_+^* in the dual by

$$f_\varrho^*(u) = \sup_{z \geq \varrho} \{uz - f(z)\}, \qquad D_{f,\varrho}(g, \nu) = C(g) + \nu + \alpha \mathbb{E}_q f_\varrho^*((g - \nu)/\alpha).$$

The same Fenchel gap decomposition holds for feasible v; strong duality still requires the corresponding attainment and regularity assumptions. Where the differentiable strictly convex response is defined, it is $v = \max\{\varrho, (f')^{-1}((g-\nu)/\alpha)\}$, with ν solved inside the response to enforce normalization. Rescaling a non-KL response afterward is generally invalid. For lower-CVaR plus a linear utility, the finite-bank reward-knot argument remains valid because the reference floor and divergence penalty do not depend on the threshold. Arbitrary nonconcave utilities do not acquire a global guarantee from this change.

What the path projection does not guarantee. Even the unrestricted mean fit need not realize the terminal target. The tilt generally changes the source law to $dS_{\lambda,0}/dQ_0 = (h_0 + \lambda)/(1+\lambda)$, whereas the model keeps Q_0. The KL chain rule therefore retains the source discrepancy $\mathrm{D}_{\mathrm{KL}}(S_{\lambda,0} \| Q_0)$ for every θ. Furthermore, tilted conditional transitions need not be Gaussian with covariance $\sigma_t^2 I$; matching their means leaves a projection error. Finite-bank ratios, sampled stored times, restricted neural predictors and changed model occupancy add further errors. Uniform time subsampling and coordinate averaging scale the population risk by fixed positive constants; arbitrary time weights do not preserve the path-KL identity. These limitations prevent substitution of this projection for the exact realization assumed by Corollary 4.

B Experimental Details

B.1 Target calibration and model fitting

FTFC assigns masses to cached pretrained samples and freezes them during fitting. Direct calibration and the normalized neural-score solver in Appendix A.2 use the same procedure.

Target calibration. Each endpoint stores a reward, utility features, and a conditioning group. Features are normalized on the training bank, while prompt or molecular-size probabilities are preserved. For concave utilities, the primal–dual gap bounds finite-bank optimization error; an independent rings solver gives a maximum gap below 2.5×10^{-6}. Lower-CVaR targets use global search, since stationarity alone is insufficient. Held-out normalization, effective sample size, and maximum density ratios assess bank coverage but do not bound population or fitting error.

Constant adapter. For a frozen linear map Wx, the adapter takes the form

$$Wx + U\big[(Vx) \odot (\gamma(c) - \gamma(0))\big], \qquad c = [1]. \tag{44}$$

Here U, V are low-rank projections and γ is a descriptor encoder. Initializing $U = 0$ recovers the pretrained predictor. The descriptor is constant, so each adapter represents one target distribution. We use AdamW with zero weight decay; ranks and update budgets are given below.

Molecular fitting. We sample complete molecules by their target masses, preserving coordinates, atom types, formal charges, and bonds. FlowMol 1 uses centered coordinates and argmax categorical labels with coordinate and cross-entropy losses. FlowMol3 retains its original time and modality weights, categorical masks, fake atoms, and stochastic self-conditioning. The loss is averaged per molecule, so size does not change its target weight. Fitting uses fresh priors and the native objective, without reward-gradient or validity losses. At generation time, atom counts follow the released training histogram; the distinction from a bank restricted to reward-eligible molecules is discussed in Appendix A.2.

Stable Diffusion fitting. We cache final latents and prompt embeddings and train with fresh diffusion noise and timesteps. For classifier-free guidance $g = 7.5$, the effective noise predictor is

$$\epsilon_{\mathrm{eff},\theta} = \epsilon_{\mathrm{pre,uncond}} + g(\epsilon_{\theta,cond} - \epsilon_{\mathrm{pre,uncond}}). \tag{45}$$

The unconditional prediction is frozen. We train on cached latents with mean-squared error against the scheduler's noise target and the same guidance convention as sampling; latents are not decoded and re-encoded.

Pretrained molecular checkpoints. QM9 fine-tuning starts from the released FlowMol 1 weights (Dunn & Koes, 2024).[1] GEOM-Drugs fine-tuning starts from the released FlowMol3 v3.0 weights (Dunn & Koes, 2025).[2]

B.2 QM9 BENCHMARK PROTOCOL

Dataset and pretrained model. QM9 contains 133,885 organic molecules with at most nine heavy atoms from C, N, O, and F, plus explicit hydrogens (Ramakrishnan et al., 2014). All methods start from the released FlowMol 1 checkpoint (Dunn & Koes, 2024) (Appendix B), which jointly models coordinates, atom types, formal charges, and bonds. We retain its preprocessing, Gaussian priors, feature schedules, categorical decoding, and atom-count histogram; fine-tuning uses generated endpoints, not the dataset's DFT labels.

Energy reward. The reward is $R(x) = -E_{\mathrm{GFN1\text{-}xTB}}(x)$ in Hartree, evaluated at the generated, unrelaxed coordinates with the predicted total charge. The differentiable `dxtb` oracle (Friede et al., 2024) supplies coordinate gradients; atom, charge, and bond labels are decoded by argmax. We use float32, implicit SCF differentiation, and tolerance 10^{-4}. Failed calculations receive -500 Ha; no draw is replaced, optimized, or fragment-pruned before evaluation.

Baseline training. AM (Domingo i Enrich et al., 2025) uses the mean-reward derivative scaled by $1/\alpha$, with $\alpha = 0.0045$, for 240 updates. FDC (De Santi et al., 2025) targets the upper superquantile at $\beta = 0.998$, with no fixed terminal KL penalty. Its proximal coefficient $\eta = 0.01$ is distinct from FTFC's terminal KL coefficient. The update counts and FDC coefficients follow Appendix E.4 of De Santi et al. (2025); Table 6 gives the remaining settings of our reconstruction.

Table 6: **Fixed QM9 baseline settings.** FTFC settings are listed in Appendix B.2.1.

Setting	AM	FDC
Objective	Mean reward, $\alpha = 0.0045$	Upper SQ, $\beta = 0.998$, $\alpha = 0$
Optimizer	Adam	Adam
Adam moments / epsilon	$(0.9, 0.999)$ / 10^{-8}	$(0.9, 0.999)$ / 10^{-8}
Learning rate	10^{-4}	10^{-4}
Optimizer updates	240	$10 \times 2 = 20$
Outer iterations	1	10
Proximal coefficient η	—	0.01
Trajectories per update	32	Up to 8 tail + 24 body
Threshold bank per outer iteration	—	4,096 fresh samples
Candidate pool per inner update	—	4,096 fresh samples
Gradient-norm clipping	10^5	10^5
Trajectory discretization	250 steps	250 steps
Training sampler	Memoryless, terminal denoising	Memoryless, terminal denoising
Evaluated checkpoint	Final scheduled update	Final outer iteration

At each outer iteration, FDC estimates the 0.998-quantile from 4,096 fresh samples and holds it fixed for two inner updates. Each inner update generates a separate pool of 4,096 trajectories and uniformly retains up to eight tail and 24 body examples. For a stratum with N_h pool members and n_h retained trajectories, each retained example has weight $BN_h/(4096n_h)$, where B is the retained batch size. This corrects the batch mean for stratified sampling. Empty tail strata contribute no examples. The reference model and inner optimizer are updated and reset, respectively, at each outer iteration.

TFFT and tail-gradient availability. TFFT uses the AM solver for 240 updates, batch size 32, learning rate 10^{-4}, and $\alpha = 0.0045$. Its threshold schedule is calibrated on 10,000 independent pretrained samples. The right-tail level rises from $\beta = 0$ to 0.998 during the first 120 updates and then

[1] Pretrained FlowMol QM9 weights: https://bits.csb.pitt.edu/files/FlowMol/trained_models_v02/qm9_gaussian/checkpoints/last.ckpt.

[2] Pretrained FlowMol3 GEOM-Drugs weights: https://bits.csb.pitt.edu/files/FlowMol/trained_models_v3/flowmol3/checkpoints/last.ckpt.

remains fixed. In the completed run, every positive-β update had zero active tail-reward gradients; only the initial expected-reward update had an active signal. We report this checkpoint, which does not demonstrate sustained tail optimization.

B.2.1 FTFC CONFIGURATION

Each of three FTFC runs uses 4,096 pretrained endpoints and upper superquantile level $\beta = 0.998$. Calibration uses rewards $-E_{\mathrm{GFN1}}/100$ and KL coefficient $\alpha = 0.01$, equivalent to one Hartree in unscaled units. It preserves the atom-count marginal and solves for the shared threshold and group normalizers to a finite-bank primal–dual gap of 10^{-8}.

With these masses fixed, we train rank-16 adapters for 80 AdamW updates: batch size 32, microbatch size 16, learning rate 10^{-3}, gradient-norm clipping at 10, and float32 precision. Sampling endpoints according to their target masses implements the weighted objective; the loss is not weighted again. We retain the native coordinate and categorical losses described in Appendix B.1. Evaluation uses the final adapter weights, without EMA or checkpoint selection by test reward. The finite-bank certificate bounds target optimization error, not fitting error.

Sampling and repeated runs. Evaluation uses 250 explicit Euler steps on a uniform grid from 0.02 to 1, with 50,000 raw attempts per run in batches of 256. All methods share the evaluation atom-count list and random draws. The table reports three FTFC runs and one run each for pretrained, AM, FDC, and TFFT, using final scheduled checkpoints; additional AM runs share random streams and are not independent repetitions.

Training time. Table 1 reports wall-clock time per run on a shared RTX 4090 server, including method-specific sample acquisition, scoring, and fitting. AM and FDC take approximately 12.0 and 35.2 hours. FTFC takes 58.4 minutes on average: $\approx 3,462.0$ seconds for bank acquisition, 0.3 seconds for calibration, and 40.5 seconds for fitting. The acquisition cost is charged even when the bank is reused. TFFT takes approximately 15 hours, including its $10,000$-sample threshold calibration. Backbone pretraining, evaluation, and development runs are excluded. These figures compare the stated budgets under shared-server load, not time to a matched reward.

B.2.2 EVALUATION METRICS

Table 1 uses all raw attempts, including the failure penalty; all reported attempts have successful energy calculations. For ascending rewards $R_{(1)} \le \cdots \le R_{(N)}$, the exact empirical superquantile is

$$\widehat{\mathrm{SQ}}_\beta = \frac{\sum_{j=1}^{k} R_{(N-j+1)} + \delta R_{(N-k)}}{m}, \qquad m = (1-\beta)N, \quad k = \lfloor m \rfloor, \quad \delta = m - k, \tag{46}$$

with the boundary term omitted when $\delta = 0$. At $N = 50{,}000$, the $\mathrm{SQ}_{0.998}$ and upper-10% columns average the best 100 and 5,000 rewards, respectively.

Graph validity requires RDKit sanitization, a single connected component, and successful canonical-SMILES conversion. It is reported as a percentage of all attempts. Synthetic accessibility (SA) (Ertl & Schuffenhauer, 2009) is averaged over valid molecules with a successful score, after removing explicit hydrogens; lower is better. FTFC entries report the mean and sample standard deviation across three runs. Baseline entries are individual-run measurements.

Topology-based validity reward. We also evaluate $R_{\mathrm{valid}} \in [0, 1]$, adapted from Kotani (2026, Eq. 13), using Eq. (49) and the GFN2-xTB protocol in Appendix B.3.2. Generated hydrogens and formal charges are retained. For molecules without hydrogens, a single-point energy replaces the hydrogen-relaxation reference. Every raw draw contributes, with zero assigned to structural or numerical failures; RDKit is not a prefilter. This metric is evaluation-only and differs from both graph validity and the GFN1 training reward.

The topology evaluation regenerates the original sampling populations from the final checkpoints. Canonical SMILES, graph-validity flags, and available geometric checks agree with the original evaluations, but missing coordinates prevent exact identity checks for every geometry.

Reproduction scope. All comparisons use this protocol and a shared checkpoint. FDC's public description omits some molecular sampling conventions, so our reconstruction has not been verified against the authors' implementation or published results.

B.3 GEOM-DRUGS BENCHMARK PROTOCOL

Pretrained model. We reconstruct the molecular task of Wang et al. (2026) using the released FlowMol3 v3.0 checkpoint (Dunn & Koes, 2025), pretrained on GEOM-Drugs with explicit hydrogens and

Kekulized bonds (Axelrod & Gómez-Bombarelli, 2022). All methods retain its atom-count histogram, coordinate dynamics, categorical continuous-time Markov chains (CTMCs), self-conditioning, and fake-atom handling. The results are computed with this shared implementation rather than taken from published tables.

Energy reward. Tables report $R(x) = -E_{\mathrm{GFN1}}(x)$ in atomic units; training uses $6R(x)$ and $\alpha = 1$, following Wang et al. (2026). We compute energies and coordinate derivatives with `dxtb` (Friede et al., 2024) in float64, using the predicted total formal charge and excluding fake atoms. Calculations allow 300 SCF iterations at tolerance 10^{-8} and one retry. Invalid molecules or failed calculations contribute no reward derivative and are ineligible for threshold estimation. They remain in the evaluation population. We apply neither energy clipping nor terminal-gradient clipping.

Baseline training. Table 7 summarizes the settings. EXP-FT uses Adjoint Matching (Domingo i Enrich et al., 2025) with expected reward; R-TFFT uses the same solver with a hinge reward. FDC (De Santi et al., 2025) uses three successive 120-update stages and two additional threshold-sampling stages. Evaluation uses the final scheduled checkpoints. The methods therefore differ in total compute.

Table 7: **Fixed molecular benchmark settings.** Method-specific choices and evaluation definitions are detailed in Appendix B.3.

Setting	Value
Runs per method	3
Reward / KL coefficient	$6(-E_{\mathrm{GFN1}})$, $\alpha = 1$
Target right-tail level	$\beta = 0.9$
Optimizer	AdamW; learning rate 10^{-4}; weight decay 0
Updates	EXP-FT / R-TFFT: 120; FDC: 3×120
Effective batch / microbatch	8 / at most 2 molecules
Training time grid	100 uniform transitions on $[0, 1]$
Adjoint interval / loss records	$t \geq 0.5$; 10 selected records
Gradient clipping	None (optimizer and terminal reward gradients)
R-TFFT curriculum	$\beta_j = 0.9j/119$, $j = 0, \ldots, 119$
Shared prior corpus	10,000 draws; 9,988 eligible
Moving FDC corpora	10,000 draws before each of stages 2 and 3
FDC stage coefficients / score time	$(1.2, 3.1, 5.0)$; $t = 0.98$
Evaluation per run	2,000 draws; batch size 16
Evaluation / corpus sampler	Native Euler/CTMC; 250 time-grid points
CTMC sampling parameters	Stochasticity 30; high-confidence threshold 0.9
Energy precision / SCF	Float64; 300 iterations; tolerance 10^{-8}

Training uses 100 uniform memoryless transitions with coordinate noise centered per molecule. The adjoint runs over $t \geq 0.5$; each loss uses the five records nearest the endpoint and five sampled from the remaining eligible records. Self-conditioning contexts are detached during the adjoint. EXP-FT and R-TFFT use residual weights $(3, 0.4, 1, 2)$ for coordinates, atom types, charges, and bonds; FDC uses only the coordinate residual (weight 1).

FDC thresholds and regularization. The first stage uses the empirical 0.9-quantile of the shared pretrained corpus; each later stage refreshes it from 10,000 samples of the preceding model. With $\eta = (1.2, 3.1, 5.0)$, the reward derivative is the hinge gradient scaled by $1/[\eta_k(1-\beta)]$, and the density regularizer is $-\alpha/\eta_k$ times the estimated current-to-pretrained coordinate-score difference. Scores use endpoint predictions at $t = 0.98$ without self-conditioning. The pretrained anchor is fixed; each stage freezes its reference model and carries over AdamW moments. We have not verified these settings against the authors' molecular implementation.

R-TFFT thresholds. The shared corpus contains 10,000 pretrained molecules; $M = 9{,}988$ have valid graphs and converged energies. The curriculum is $\beta_j = 0.9j/119$ for $j = 0, \ldots, 119$. For $\beta_j > 0$, solve

$$\tau_j \in \arg\min_\tau \left\{ \tau + a \log \left[\frac{1}{M} \sum_{i=1}^{M} \exp\left(\frac{[R_i - \tau]_+}{a(1-\beta_j)} \right) \right] \right\}, \qquad a = \frac{\alpha}{6} = \frac{1}{6}. \tag{47}$$

Use bounded scalar minimization with tolerance 10^{-7} on $[R_{\min} - 2s, R_{\max} + 2s]$, where $s = \max(R_{\max} - R_{\min}, \mathrm{std}(R), 1)$. At $\beta_0 = 0$, $\tau_0 = R_{\min} - 20s$ recovers the expected-reward derivative. The training pseudo-reward is $6[R(x) - \tau_j]_+/(1 - \beta_j)$.

Tail-gradient availability. For positive curriculum values, thresholds range from 176.8775 to 179.0905, but only one eligible pretrained reward exceeds them. No eligible training molecule exceeds its threshold in any R-TFFT run, so positive-β updates receive no active hinge-reward gradient. The initial expected-reward update and later reference matching can still change the model. We report all three runs; this is a limitation of our finite-sample reconstruction, not of TFFT in general.

B.3.1 Evaluation samples and metrics

Each method is evaluated in three runs with 2,000 molecules per run, using the native Euler/CTMC sampler with 250 time-grid points and batch size 16. Evaluation uses common random draws across methods. Failed or repeated molecules are retained without replacement or deduplication.

Energy and tail reward. Mean reward averages the unscaled negative GFN1 energy over molecules that pass RDKit sanitization and have a finite, converged energy. For the N_s eligible rewards in run s, ordered increasingly, the right-tail statistic is

$$\widehat{\text{R-CVaR}}_{0.9,s} = \frac{1}{m_s} \sum_{i=N_s-m_s+1}^{N_s} R_{(i)}, \qquad m_s = \lceil 0.1 N_s \rceil. \tag{48}$$

This averages the best-scoring tenth, with equal weights and no fractional boundary correction. Both energy metrics condition on successful scoring. Since they use total energies of molecules with varying size and composition, higher reward alone does not establish better geometry or synthesizability.

Graph validity and synthetic accessibility. Validity is the percentage of all draws passing RDKit sanitization. It does not require connectivity, energy convergence, or a stable three-dimensional geometry. SA (Ertl & Schuffenhauer, 2009) is averaged over sanitized molecules with finite scores, regardless of energy convergence; lower is better. The evaluation retains explicit hydrogens when computing SA. This differs from the QM9 convention and must be accounted for when comparing SA across benchmarks.

Aggregation. Tables report the mean and sample standard deviation across three runs, computed from each run's metric. Validity uncertainty is in percentage points. The runs share one pretrained corpus, so this variation does not include uncertainty from acquiring a new corpus. Energy and SA use the eligible populations above; the topology reward below uses all generated molecules.

Qualitative examples. Figure 4 shows four molecules per method. The baselines use the first four draws from a separate sampling run. FTFC uses an earlier checkpoint than the quantitative table: three panels are unselected draws, while Draw 2 is the first new valid molecule with converged reward $R \geq 80$. The first candidate met this threshold. This selected panel is excluded from quantitative evaluation and cannot be used to compare overall performance.

All methods use the same sampling settings as the benchmark. The annotations give the unscaled GFN1 rewards at the generated geometries. Rendering applies only rigid alignment, hides hydrogens, and fits each molecule to its panel; it does not optimize coordinates. Panel sizes therefore do not represent a common spatial scale.

B.3.2 Topology-based validity reward

We adapt the validity reward of Kotani (2026, Eq. 13) to measure whether relaxation preserves a generated molecule's heavy-atom connectivity and how much energy it releases. All 30,000 benchmark draws are scored, including RDKit-rejected graphs and failed GFN1 calculations. This metric is used only for evaluation.

Definition. Let A_i indicate that molecule i passes the connectivity, collision, relaxation, and hydrogen-integrity checks below. We call this event xTB topology preservation (XTP). Let h_i be its number of heavy atoms, E_i^{H} the energy after optimizing hydrogens with heavy atoms fixed, and E_i^{F} the energy after unconstrained relaxation. With energies in kcal/mol, the reward is

$$v_i = \begin{cases} 0, & A_i = 0, \\ 0.60 + 0.40 \exp(-\varepsilon_i / T_\varepsilon), & A_i = 1, \end{cases} \qquad \varepsilon_i = \frac{|E_i^{\mathrm{H}} - E_i^{\mathrm{F}}|}{h_i}, \quad T_\varepsilon = 2.00. \tag{49}$$

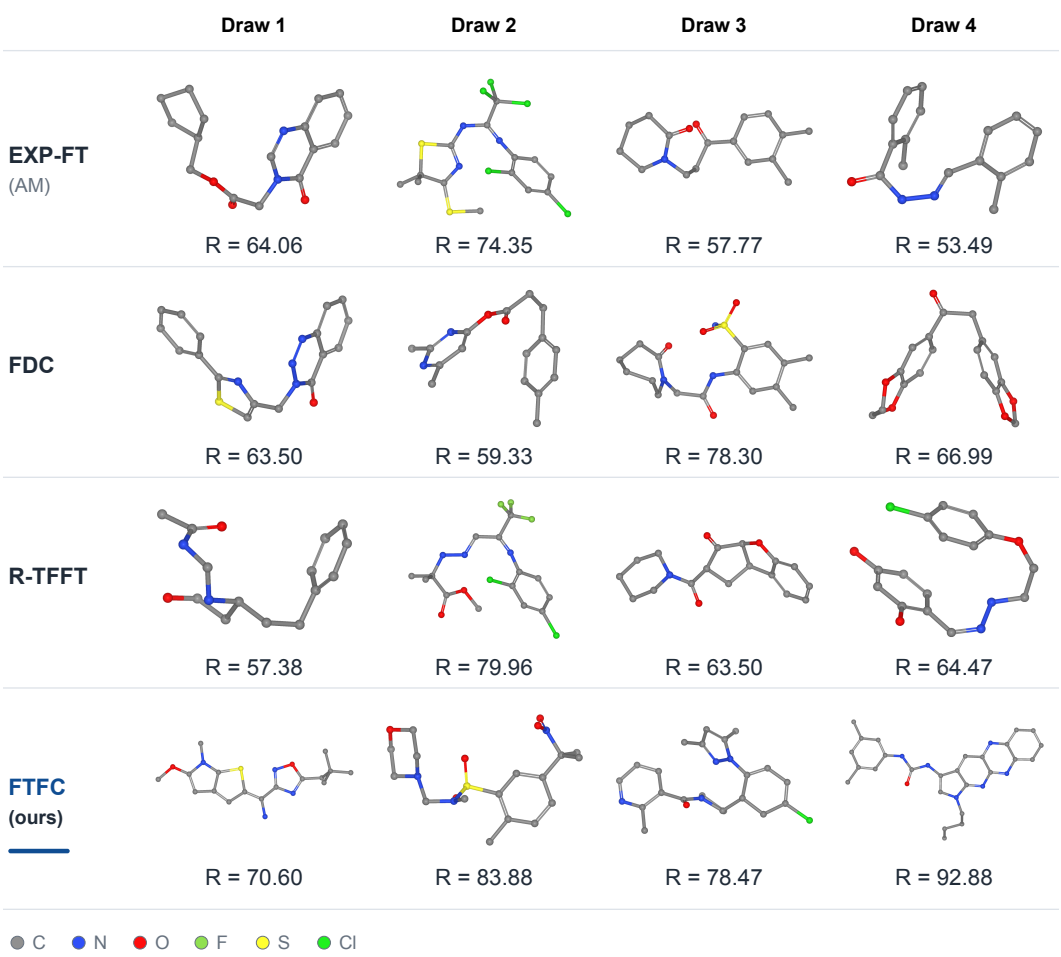

Figure 4: **Molecular examples with one threshold-selected panel.** FTFC Draw 2 is the first new sample with measured reward $R \geq 80$; the other panels show unselected draws. This selection is illustrative and does not establish comparative performance. Labels give the measured, unscaled GFN1 reward $R = -E_{\mathrm{GFN1}}$. FTFC uses an earlier checkpoint. The rendering preserves generated geometry, hides hydrogens, and fits each view to its panel. Appendix B.3.1 specifies the protocol and number of candidates.

The strain ε_i and scale T_ε are in kcal/mol per heavy atom. The absolute energy gap follows the released reward implementation and agrees with the signed relaxation decrease for all passing samples here. Passing molecules receive a base reward of 0.60, with up to 0.40 more for low strain; any structural or numerical failure receives zero.

Relaxation protocol. We retain generated hydrogens and formal charges, remove fake atoms, and choose the minimum spin consistent with electron-count parity. The reference heavy-atom graph uses predicted bonds without bond orders; RDKit sanitization is not a prefilter. After connectivity and the released ADT collision checks, GFN2-xTB relaxation proceeds in three stages:

1. Optimize hydrogens with all heavy atoms fixed, using L-BFGS for at most 500 cycles.
2. Relax all atoms while restraining the predicted heavy-atom bond distances to their values after hydrogen relaxation, with force constant 0.5.
3. Remove all restraints and perform unconstrained relaxation.

Every stage must converge with finite coordinates and energy. We use xTB 6.7.1, accuracy 1.0, at most 250 SCC iterations, and standard geometry optimization tolerances. Time limits are 300 seconds for hydrogen relaxation and 180 seconds for each later stage; timeouts and numerical failures score zero.

In the final geometry, atoms i, j are connected when $\|x_i - x_j\| < 1.3[r_{\text{cov}}(Z_i) + r_{\text{cov}}(Z_j)]$, using Cordero covalent radii. The resulting heavy-atom edge set must match the predicted graph exactly. Hydrogens farther than 1.6 Å from every heavy atom are detached. An even number of detached hydrogens is removed and relaxation restarts, up to three times; an odd number causes rejection. These rules—native hydrogens, rejection of odd detachments, and convergence at every stage—differ from the authors' full generation and scoring pipeline.

Aggregation and interpretation. For $n = 2{,}000$ molecules in run s, define

$$R_{\text{valid},s} = \frac{1}{n}\sum_{i=1}^{n} v_{s,i}, \qquad p_s = \frac{1}{n}\sum_{i=1}^{n} A_{s,i}, \qquad q_s = \frac{\sum_{i:A_{s,i}=1} v_{s,i}}{\sum_i A_{s,i}}. \tag{50}$$

Here p_s is the XTP pass rate and q_s the conditional mean reward. When $p_s > 0$,

$$R_{\text{valid},s} = p_s q_s, \qquad 0.60 p_s \le R_{\text{valid},s} \le p_s \le 1. \tag{51}$$

Failures remain in the denominator, so R_{valid} combines topology preservation with the strain of passing molecules; it is not a validity percentage. It also differs from the GFN1 training reward. Local connectivity preservation does not establish a global energy minimum or agreement with a reference conformer.

Table 8 separates these effects. FTFC has the highest observed pass rate, while pretrained has the lowest relaxation strain. R-TFFT passes more often than EXP-FT and FDC but has greater strain among passing molecules; graph validity or XTP rate alone is therefore insufficient to characterize the combined reward.

Table 8: **Decomposition of the additional validity reward.** These diagnostics cover the same 30,000 draws as Table 3. XTP is the fraction of all draws that pass the complete protocol. The remaining columns condition on XTP: mean validity reward and mean relaxation strain in kcal/mol per heavy atom. Each entry is the mean ± sample standard deviation of three runs. Bold values mark the best means before rounding: higher XTP and reward, lower strain.

Method	XTP (%)	Reward given XTP	Strain given XTP
Pretrained	98.52 ±0.20	**0.97** ±0.00	**0.18** ±0.00
EXP-FT (AM)	97.68 ±0.12	0.95 ±0.00	0.27 ±0.02
FDC ($K = 3$)	97.68 ±1.08	0.95 ±0.01	0.27 ±0.04
R-TFFT	98.23 ±0.26	0.94 ±0.02	0.32 ±0.11
FTFC (ours)	**98.67** ±0.08	0.97 ±0.00	0.19 ±0.00

B.3.3 FTFC CONFIGURATION AND TRAINING COST

FTFC training. FTFC solves the upper-CVaR dual with $\beta = 0.9$, KL coefficient $\alpha = 1$, and reward scale six on the 9,988 eligible pretrained endpoints. Normalization preserves the empirical masses of 92 molecular size groups. The fixed target is fitted with a rank-16 adapter for 120 updates: batch size eight, microbatch size one, learning rate 10^{-4}, float32 precision, and gradient-norm clipping at ten. No new reward queries are needed during fitting. Evaluation uses the final adapter weights.

Cost with a cached corpus. Table 9 reports active GPU-allocation time per run on an RTX 3090 Ti, including CPU work within training. It includes setup, calibration, fitting, and method-specific online sampling and scoring. The shared pretrained corpus is already available; evaluation, backbone pretraining, and interruption downtime are excluded. FTFC calibration takes 0.246 seconds and fitting 132.52 seconds on average, or 2.36 minutes including setup and export. FDC additionally requires two 10,000-molecule threshold refreshes. These timings compare the stated training budgets, rather than time to reach the same reward; the R-TFFT comparison also has the tail-gradient limitation described in Appendix B.3.

B.4 STABLE DIFFUSION BENCHMARK PROTOCOL

Model and prompts. All methods start from Stable Diffusion v1.5. Baselines use the released TFFT implementation,[3] which updates the full U-Net while freezing the VAE, text encoder, reward model, and pretrained reference. FTFC uses the adapter procedure in Appendix B.1.

[3] https://github.com/zifan-5528/TFFT

Table 9: **Training time on GEOM-Drugs.** Active minutes per run on one RTX 3090 Ti, with the shared pretrained corpus available. Values are mean ± sample standard deviation over three runs. Bold values mark the lowest mean times among fine-tuned methods, including ties; updates are prescribed budgets.

Method	Updates	Fitting + setup	Threshold refresh	Total training
Pretrained	0	0.00	0.00	0.00
EXP-FT (AM)	120	129.40 ±0.89	**0.00**	129.40 ±0.89
FDC ($K = 3$)	360	388.11 ±2.07	414.65 ±11.03	802.75 ±13.02
R-TFFT	120	125.83 ±0.36	**0.00**	125.83 ±0.36
FTFC (ours)	120	**2.36** ±0.08	**0.00**	**2.36** ±0.08

Evaluation, original backbone pretraining, and documented interruption intervals are excluded. FDC's two required 10,000-molecule threshold refreshes are included. The common offline corpus is accounted for separately in the text. Total uncertainty is computed from per-run totals; rounded component means need not sum exactly to the rounded total.

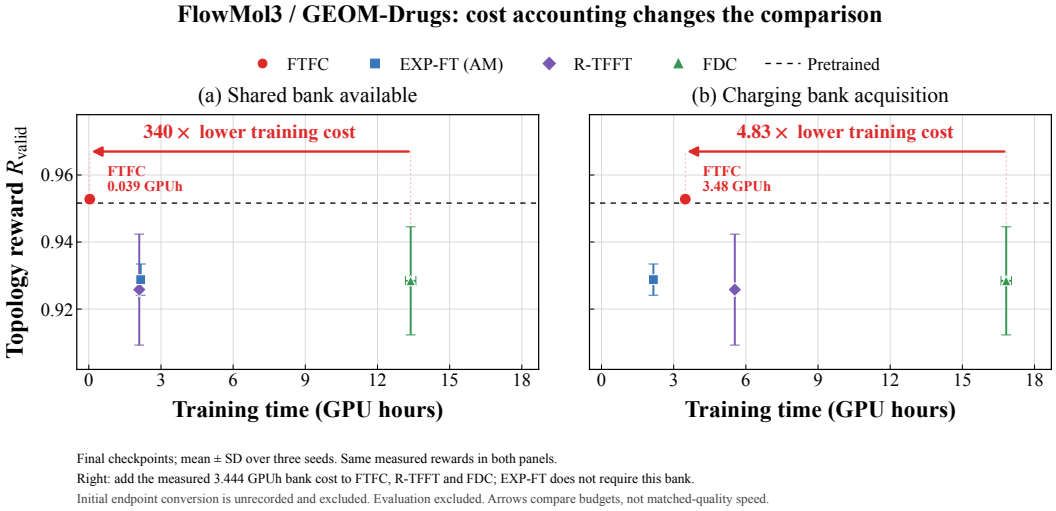

Figure 5: **Effect of corpus acquisition on GEOM-Drugs training cost.** Both panels use the same final topology rewards, averaged over three runs with 2,000 generated molecules per run; error bars show sample standard deviations. (a) The shared corpus is available. (b) Its measured acquisition cost, 3.444 GPU-hours, is charged once to each method that requires it. Initial endpoint conversion was not timed and is excluded, as is evaluation. Arrows compare recorded budgets rather than time to a matched reward. The horizontal dashed line shows the pretrained topology reward.

The training list has 10,000 records and 9,803 distinct prompts; evaluation uses 100 prompts, two of which also occur in training, including the astronaut prompt in Figure 3. Baselines use their final scheduled checkpoints. FTFC uses the evaluated checkpoint with the highest lower-tail CVaR, so its result includes evaluation-set selection.

Baseline objectives and budgets. For raw ImageReward $r(x, c)$, EXP-FT uses Adjoint Matching (Domingo i Enrich et al., 2025) with the identity reward. L-TFFT and our two-stage FDC reconstruction use a lower-tail hinge:

$$\phi_{\mathrm{EXP}}(r) = r, \qquad \phi_\tau(r) = -\frac{[\tau - r]_+}{\beta}, \qquad \beta = 0.2, \quad [a]_+ = \max(a, 0). \tag{52}$$

Table 10 gives thresholds and budgets. Thresholds are fixed within each stage. FDC's second stage continues from the first with a fresh optimizer and warmup. These settings reconstruct the baselines (De Santi et al., 2025; Wang et al., 2026); exact agreement with the authors' experiments has not been established.

Table 10: Stable Diffusion method budgets and training thresholds. All trainable baselines are assigned 1,600 optimizer updates in total.

Method	Epochs	Optimizer updates	Raw hinge threshold τ
Pretrained (Rombach et al., 2022)	0	0	—
EXP-FT (Domingo i Enrich et al., 2025)	20	1,600	Identity reward
L-TFFT (Wang et al., 2026)	20	1,600	0.700759
FDC, stage 1 (De Santi et al., 2025)	10	800	-0.49
FDC, stage 2 (De Santi et al., 2025)	10	800	-0.09

Gradient estimation and optimization. The terminal adjoint is $-100g$. For each endpoint, we add 20 Gaussian perturbations with standard deviation 0.02, decode and score them, and differentiate the identity or hinge reward. Norms are clipped at the per-image 85th percentile before averaging, so the hinge acts before smoothing and clipping. The control penalty remains relative to the frozen pretrained U-Net. Table 11 gives the remaining settings: each epoch has 80 updates with 125 replay records per update; each loss minibatch samples two denoising positions from the first 60% of the trajectory and two from the remainder. Replay losses are not importance-weighted. Guidance is used only for image sampling, not adjoint or control computations.

Table 11: Shared Stable Diffusion baseline training settings. Reward-gradient clipping during smoothing is distinct from optimizer-gradient clipping.

Setting	Value
Optimizer	AdamW; learning rate 3×10^{-6}
Adam moments, epsilon	$(0.9, 0.95)$, 10^{-8}
Weight decay	0
Learning-rate schedule	Linear warmup for 20 updates, then constant
Minibatch / accumulation	5 records / 25 minibatches
Replay buffer / passes	100 trajectories / 10 passes
Trajectory / loss positions	50 denoising steps / 4 sampled positions
Training reward	ImageReward-v1.0; multiplier 100
Tail objective	Lower tail, $\beta = 0.2$, for FDC and L-TFFT
Gradient smoothing	20 latent perturbations, Gaussian std. 0.02
Smoothing gradient clipping	Per-image 0.85-quantile of perturbation norms
Other clipping	Optimizer-gradient and per-sample loss clipping disabled
Guidance / prompt dropout	Sampling CFG 7.5 / dropout 0.2
Adjoint / control CFG	Both disabled
Precision	Float32; TF32 permitted
Validation frequency	Every 0.1 epoch on the 100 evaluation prompts
Final model selection	Last checkpoint after the scheduled updates
Hardware per worker	One NVIDIA A100-SXM4 with 80 GB memory

Sampling. Training and evaluation use the released stochastic DDIM scheduler with 50 steps, $\eta = 1$, trailing timestep spacing, and noise-schedule endpoints 0.002 and 0.009. The terminal cumulative alpha is adjusted to $(1 + \bar{\alpha}_{t_{\mathrm{last}}})/2$. Each method generates ten 512×512 images for

each of 100 prompts, using guidance 7.5 and common random draws across methods. All 1,000 images enter quantitative evaluation without selection. Scorers use their released preprocessing and unperturbed images.

ImageReward and lower-tail CVaR. ImageReward-v1.0 (Xu et al., 2023) predicts preference for a prompt–image pair. Let r_{pi} be its raw score, $P = 100$ the number of prompts, and $n = 10$ the number of images per prompt. The mean is

$$\widehat{\mu}_{\mathrm{IR}} = \frac{1}{P}\sum_{p=0}^{P-1} \bar{r}_p, \qquad \bar{r}_p = \frac{1}{n}\sum_{i=1}^{n} r_{pi}. \tag{53}$$

Sorting all $M = Pn$ scores jointly gives the lower-tail statistic

$$\widehat{\text{L-CVaR}}_{0.2} = \frac{1}{K}\sum_{j=1}^{K} r_{(j)}, \qquad K = \lceil 0.2M \rceil = 200. \tag{54}$$

This is a global tail across prompts, not an average of per-prompt tails. Evaluation uses raw scores and an empirical tail boundary, independently of the fixed training thresholds.

CLIP and HPSv2.1. CLIP ViT-L/14 (Radford et al., 2021) measures text–image compatibility through the cosine similarity of image and text features:

$$C(x, c) = \frac{f(x)^\top h(c)}{\|f(x)\|_2 \|h(c)\|_2}. \tag{55}$$

HPSv2.1 (Wu et al., 2023) supplies a separate learned preference score. Both are reported without rescaling, averaged within prompts and then across prompts. Neither is used for training.

DreamSim diversity. For unit-normalized embeddings z_{pi} from the default pretrained DreamSim model (Fu et al., 2023), we compute

$$D_p = \frac{1}{2n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n} \|z_{pi} - z_{pj}\|_2^2, \qquad \widehat{D} = \frac{1}{P}\sum_{p=0}^{P-1} D_p. \tag{56}$$

This is within-prompt perceptual variance, used only for evaluation; higher diversity need not imply higher image quality or prompt alignment.

Uncertainty. The table reports one evaluation per method. For ImageReward mean, CLIP, and HPS, let a_p be the ten-image prompt mean; for DreamSim, let $a_p = D_p$. The reported error term is

$$\mathrm{SE}_{\mathrm{prompt}} = \frac{\sqrt{P^{-1}\sum_{p=0}^{P-1}(a_p - \bar{a})^2}}{\sqrt{P}}, \qquad \bar{a} = P^{-1}\sum_{p} a_p. \tag{57}$$

For lower-tail CVaR, it is instead

$$\mathrm{SE}_{\mathrm{tail}} = \frac{\sqrt{K^{-1}\sum_{j=1}^{K}(r_{(j)} - \widehat{\text{L-CVaR}}_{0.2})^2}}{\sqrt{K}}, \qquad K = 200. \tag{58}$$

The tail term describes dispersion within the selected tail; it omits boundary uncertainty and dependence between images of one prompt. Neither term measures training-run variability or gives a confidence interval for method differences.

Training cost. Baseline timings on one A100 80 GB GPU include active training, validation, checkpoint writing, and repeated partial work after interruptions. They exclude downtime, external startup, final evaluation, backbone pretraining, and development runs. The recorded totals are 82.28 hours for EXP-FT, 82.41 for FDC, and 85.77 for L-TFFT.

The selected FTFC checkpoint combines 800 initial updates and a 400-update refinement. Fitting takes 4.045 GPU-hours with a cached bank. Generating and scoring its 80,000-image bank adds 43.711 GPU-hours, giving 47.756 GPU-hours (1.99 GPU-days) in total. FTFC's timers for these stages exclude CPU calibration, startup, checkpoint writing, validation, and exploratory or discarded updates. Because timing scopes and attained rewards differ, these costs do not establish a speedup to matched quality.

Qualitative selection. Figure 3 shows ImageReward ranks 1, 5, and 10 among ten images per method for "*footage of an astronaut in a tropical beach*." This prompt was chosen after inspecting pretrained outputs and also occurs in training. Pretrained, EXP-FT, L-TFFT, and FTFC reuse their evaluation images. The FDC batch was resampled twice after visual inspection; its displayed ranks use the last batch. This selection affects only the illustration, not the quantitative benchmark. No images are retouched.

B.4.1 L-TFFT: FIXED-THRESHOLD LOWER-TAIL OPTIMIZATION

L-TFFT applies the lower-tail transformation of Wang et al. (2026) with $\beta = 0.2$ and fixed threshold $\tau = 0.700759$: $\phi_\tau(r) = -5[0.700759 - r]_+$. For a terminal latent z and decoded reward $r(z, c)$, its unsmoothed derivative is

$$\nabla_z \phi_\tau(r(z, c)) = 5\, \mathbf{1}\{r(z, c) < \tau\}\, \nabla_z r(z, c), \tag{59}$$

away from the threshold. Low-reward samples receive a direct reward gradient, while the KL control penalty remains active for all samples. The threshold is fixed, so the fraction below it need not remain 20%.

The hinge is applied to each latent perturbation before clipping and averaging, so an image above the threshold can still receive a gradient when a perturbation falls below it. Training uses the shared Adjoint Matching settings and 1,600 updates in Tables 10–11.

L-TFFT improves mean ImageReward and lower-tail CVaR over pretrained, but not over EXP-FT in this evaluation. With one run per method, this difference does not establish equivalence or statistical superiority.

C Additional Results

C.1 Two-dimensional rings

Protocol and numerical targets. We retain the fixed pretrained VP network and stochastic DDIM sampler ($\eta = 1$), with the $L = 64$ pretrained law as q. The training bank contains 65,536 endpoints; the disjoint evaluation bank contains 262,144. All three objectives use terminal KL coefficient $\alpha = 0.03$. The left-tail objective in Section 4.1 is

$$\mathcal{F}(p) = \text{L-CVaR}_{0.2,p}(r) = \sup_{\tau \in \mathbb{R}} \left\{ \tau - \frac{1}{0.2} \mathbb{E}_p[(\tau - r(X))_+] \right\}.$$

For mean reward, $p_i^\star \propto q_i \exp(r_i/\alpha)$; left-CVaR uses an exhaustive finite-bank threshold solve with tail mass 0.2. For the CVaR baselines, a separate calibration bank determines the threshold $\widehat{\tau}$, and both ITM (Potaptchik et al., 2025) and FDC (De Santi et al., 2025) receive the fixed scalar reward $-(\widehat{\tau} - r(x))_+/0.2$. ITM applies exponential tilting to this reward; it is not supplied the numerical target's masses. The evaluation bank is used only for independent evaluation. Coverage entropy uses 16 softmax radial-basis features, centered at eight equally spaced points on each circle (width 0.38). Its finite-bank target minimizes the convex dual $\alpha \log \sum_i q_i \exp((r_i + \phi_i^\top b)/\alpha) + 0.12 \log \sum_k \exp(-b_k/0.12)$. We check its primal–dual gap and provide the same smooth supporting reward $r(x) + \phi(x)^\top b$ to ITM and FDC. Numerical targets approximate the population optimum; they are not trained FTFC outputs.

Training and evaluation. Hyperparameters are fixed before the three reported runs. FTFC uses a rank-32 constant adapter, AdamW with learning rate 0.003, batch size 8,192, and EMA weights. Mean reward, lower-CVaR, and coverage entropy use 256, 768, and 512 updates, respectively. We fit a separate adapter for each objective. Normalized neural scores determine the target masses, which are checked against independent finite-bank solutions. Fitting uses fresh VP noise and epsilon regression, with training times $(1 - U)^2$ quantized to 128 bins. ITM uses 8,192 updates and FDC 512, both with batch size 128; these are different training budgets.

Each run contributes 4,096 unfiltered evaluation samples. Figure 1 displays a random subset from one run, while its annotations summarize all three. The numerical target uses the independent bank of 262,144 pretrained endpoints. No samples are projected onto the rings or otherwise corrected.

Table 12: **Agreement with the objective-specific numerical target.** Sliced-Wasserstein distance (SW_1, lower is better) at $L = 64$, reported as mean $\pm$ sample standard deviation across three runs. Each run contributes all 4,096 raw generated samples. The lowest mean in each row is **bold**.

Objective	FTFC	ITM	FDC
Mean reward	**0.039** $\pm$0.009	0.095 $\pm$0.027	0.287 $\pm$0.006
Left-CVaR	**0.039** $\pm$0.015	0.121 $\pm$0.005	0.350 $\pm$0.007
Coverage entropy	**0.032** $\pm$0.008	0.205 $\pm$0.080	0.892 $\pm$0.066

Sliced-Wasserstein distance assesses relative mode masses and angular coverage as well as proximity to the rings. FTFC has the lowest mean distance for all three objectives. These distances are measured against finite-bank numerical targets; the reported standard deviations summarize only these three runs.

C.1.1 MEAN REWARD

Mean reward favors either circle, while the KL penalty retains the reference law's preference for the left ring. FTFC has the smallest target distance at $L = 64$ (Table 12), reproducing both modes and their unequal masses more closely than ITM or FDC.

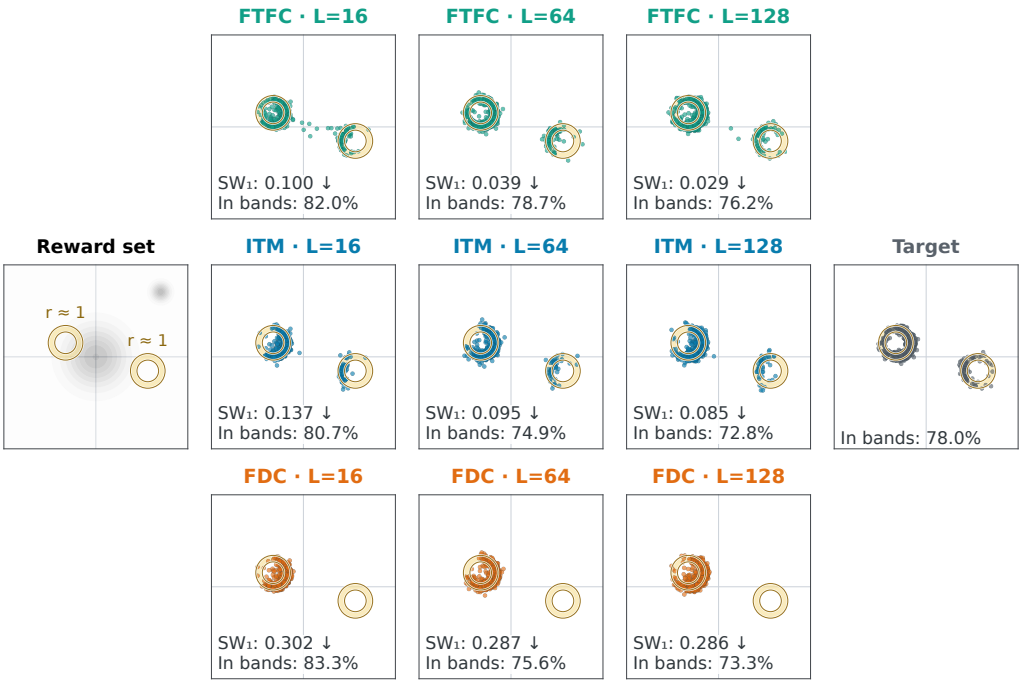

Figure 6: **Mean-reward objective**, $\mathcal{F}(p) = \mathbb{E}_p[r]$. The target is computed independently. Rows show FTFC, ITM, and FDC at $L = 16, 64, 128$; the target is at middle right. Each panel shows 500 samples from one run; annotations use all 4,096 samples across three runs.

Convergence and cost. Figure 7 uses the same runs and numerical targets.

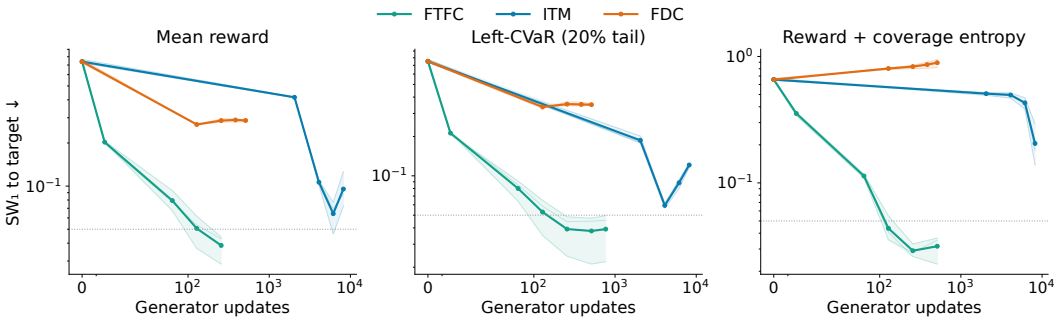

Figure 7: **Convergence to the numerical target at $L = 64$.** Solid curves average three runs and shading spans their range. The dotted line marks sliced $\mathrm{W1} = 0.05$. The axis counts generator updates; target optimization and endpoint acquisition are excluded. Batch sizes and sampling costs differ, so the curves do not establish equal-compute or end-to-end wall-clock speedups.

C.1.2 COVERAGE ENTROPY

The coverage objective $\mathcal{F}(p) = \mathbb{E}_p[r] + 0.12H(\mathbb{E}_p[\phi])$ rewards entropy of aggregate feature masses, not the entropy of an individual sample. It encourages both circles and their arcs (Figure 8). FTFC most closely matches this target; ITM concentrates on arcs and FDC largely misses the right ring.

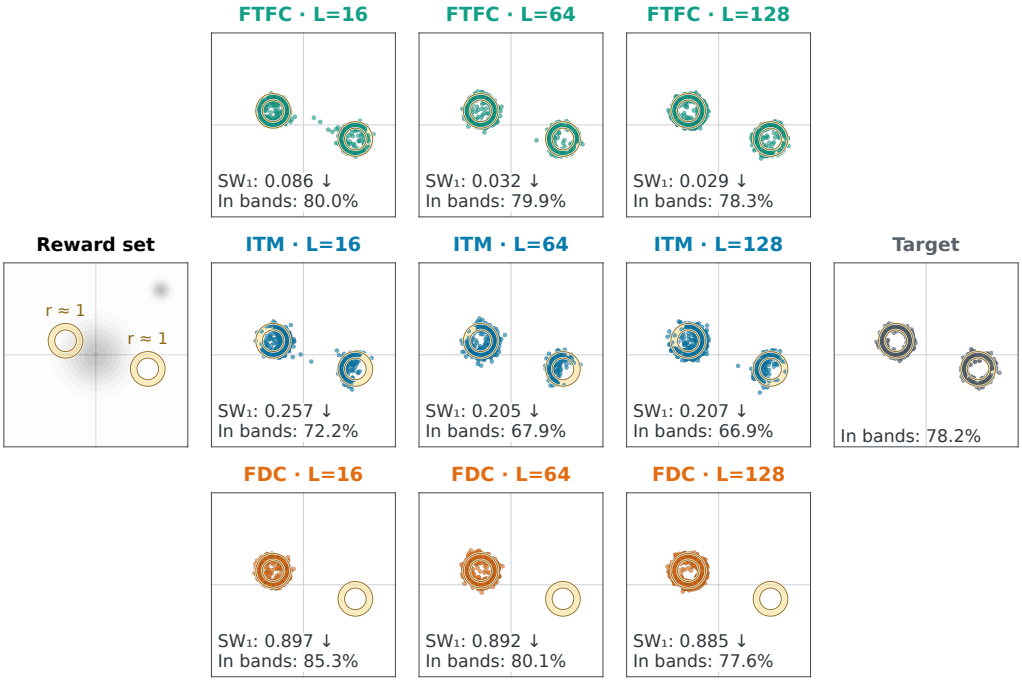

Figure 8: **Coverage-entropy objective.** $\mathcal{F}(p) = \mathbb{E}_p[r]+0.12H(\mathbb{E}_p[\phi])$ with KL coefficient $\alpha = 0.03$. Rows, sampling lengths, evaluation protocol, and displayed run follow Figure 1. The target retains mass outside the gold bands, so band-hit rate alone does not measure agreement.