ON ORACLE DESIGN FOR GUIDED DIFFUSION IN THE LOW-DATA REGIME

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we propose to use the reverse dynamics of diffusion models to train oracles suited for steering in the low-data regime. Steering diffusion models toward desired properties at inference time requires oracles that properly map intermediate noised states to end-state properties. In the scarce labelled data regime, however, learning such a mapping is challenging. Crucially, the reverse process dynamics carry valuable learning signal on how predictions should evolve across timesteps, one that can be leveraged without any labels. We exploit this by training oracles in the original data space while constraining their predictions on consecutive intermediate noised states to adhere to reverse process dynamics, requiring no additional labelling. Optimizing this objective is non-trivial (its loss nests an expectation over the stochastic reverse dynamics), for which we develop an anchored-target scheme to facilitate a principled optimization. In a systematic benchmark of thousands of experimental configurations, spanning regression and classification tasks in both continuous and discrete design spaces, our approach achieves superior steering quality across varied labelled data availability. These results establish dynamics-aware oracle training as the right strategy in the label-scarce regime, broadly applicable across diffusion and flow-based models. Code is available at `https://anonymous.4open.science/r/code_dapp-0F53/`.

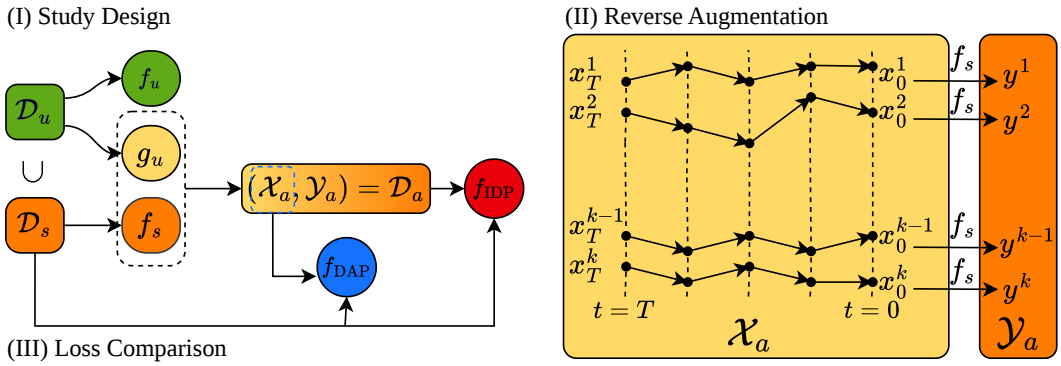
1 INTRODUCTION

Diffusion models Song et al. (2020); Ho et al. (2020) have demonstrated remarkable capacity to capture complex data distributions across diverse scientific domains. Harnessing this capacity for real-world problems, however, requires steering the generation process towards samples with desirable properties. Inference-time steering is a particularly appealing paradigm for this, as it requires no retraining of the generative model and can target any property of interest, making it broadly applicable across scientific tasks. All inference-time steering approaches Song et al. (2020); Singhal et al. (2025); Bansal et al. (2023); Chung et al. (2022); Janati et al. (2024); Trippe et al. (2022); Wu et al. (2023); Dou & Song (2024); Zhao et al. (2025); Cardoso et al. (2023); Li et al. (2024) rely on a property oracle to evaluate samples during generation, where oracle quality directly determines steering quality. Rather than proposing a new steering algorithm, we address an orthogonal question: how to train the property oracle these algorithms rely on, given scarce labelled data.

In scientific applications, labelled data is inherently scarce: measuring properties of interest, such as reaction yields for novel catalysts or binding affinities for newly synthesised molecules, requires expensive and time-consuming laboratory experiments. Yet prior work on oracle training has largely assumed access to a labelled dataset of comparable scale to the diffusion model's training set Song et al. (2020); Nisonoff et al. (2024), an assumption that rarely holds in practice.

A natural strategy to address data scarcity is to exploit the prior generative model itself. By reverse sampling from the trained diffusion model, one can generate trajectories whose intermediate states span the generation path, providing additional training signal beyond the small labelled dataset. We propose to use these augmented trajectories to train a property oracle that operates directly on intermediate noised states. The key insight is that the reverse diffusion dynamics impose a well-defined relationship between the oracle's predictions at consecutive time steps: the prediction at a noisier state x_t must equal the expected prediction at the less noisy state x_{t-1} under the posterior $p(x_{t-1}|x_t)$. Training the oracle to satisfy this inter-step constraint provides a learning signal

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.



(III) Loss Comparison

Loss	DAP	IDP
Boundary	$\mathcal{L}(y_s, f_\theta(x_s))$	$\mathcal{L}(y_s, f_\theta(x_s))$
Intermediate	$\mathcal{L}(f_\theta(x_{t-1}^i, f_\theta(x_t^i))$	$\mathcal{L}(y^i, f_\theta(x_t^i))$

- $(x_s, y_s) \in \mathcal{D}_s$
- $x_{t-1}^i, x_t^i \in \mathcal{X}_a,\ y^i \in \mathcal{Y}_a$
- $x_{t-1}^i \sim p(x_{t-1}|x_t^i)$

Figure 1: **Overview of the study framework.** (I) Dataset-model relationships: arrows indicate which datasets train which models; f_u serves as the ground truth and D_s is a small labelled subset. (II) Reverse augmentation: trajectories from g_u form $\mathcal{X}_a$; trajectory endpoints labelled by f_s form pseudo-labels $\mathcal{Y}_a$. (III) Loss comparison: DAP and IDP share the boundary loss on D_s; DAP enforces inter-step constraints on $\mathcal{X}_a$ while IDP maps intermediate states to pseudo-labels $\mathcal{Y}_a$.

derived entirely from the structure of the generative process, requiring no labels for the generated trajectory states. We call the resulting oracle the **Dynamics-Aware Predictor (DAP)**. Training DAP on this constraint is non-trivial: its loss contains a nested expectation over the stochastic reverse dynamics, and a naive single-sample estimator is systematically biased, pulling the oracle toward trivial, dynamics-smoothing solutions. We address this with an anchored-target training scheme, backed by theoretical analysis.

We conduct a systematic study comparing DAP against oracle variants that operate in the clean data space or directly map intermediate states to endpoint labels without explicitly enforcing the intermediate dynamics, across a wide range of labelled dataset sizes, augmented trajectory counts, and label quality levels. All oracles are evaluated on intermediate-state prediction quality and on their effectiveness as steering potentials under Feynman-Kac (FK) particle-based steering Singhal et al. (2025), recently adopted in protein design and structure prediction Hartman et al. (2025); Didi et al. (2026); Shen et al. (2025); Passaro et al. (2025), as well as gradient-based predictor guidance across discrete Nisonoff et al. (2024) and continuous design spaces Song et al. (2020); Dhariwal & Nichol (2021). We use controlled benchmarks of increasing data complexity and steering task difficulty: half-moon, an interpretable 2D setting; MNIST, a more diverse multi-class setting; and QMugs Isert et al. (2022), a dataset of drug-like molecules representative of real scientific design spaces; together they span classification and regression targets across continuous diffusion and discrete flow models, totalling thousands of experimental configurations. We discuss related work in Appendix A.14.

Our contributions are: **(1) DAP**: to our knowledge, the first approach to exploit reverse-trajectory augmentation for property oracle training, improving oracle training under scarce labelled data. **(2) Anchored-target training**: an anchored-target method that resolves the bias of the naive estimator, backed by theoretical analysis. **(3) Comprehensive benchmarking**: to our knowledge, the first study to investigate FK-based steering and gradient-based predictor guidance in the low-data regime, systematically evaluating oracle variants across prediction and steering quality under varying labelled dataset sizes and trajectory counts.

2 Preliminaries

We consider the problem of training a property oracle f_θ that can evaluate samples along the reverse diffusion process, given access to a small labelled dataset $D_s = \{(x_i, y_i)\}_{i=1}^{|D_s|}$ in the original space and a pretrained generative model g_u. The generative model is trained on a large universe dataset D_u,

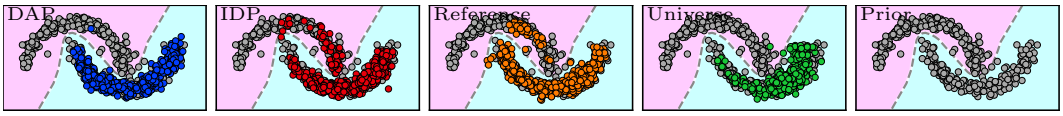


Figure 2: **DAP produces target-confined samples with broad coverage under scarce labelled data.** In contrast, the universe oracle (f_u) operating via the posterior mean produces a skewed distribution. Background: class boundaries from universe oracle f_u; grey: unguided prior samples.

Algorithm 1 Anchored-target training for DAP

Require: Initial parameters θ_0; learning rate η; EMA rate $\alpha \in (0, 1)$; intermediate-loss weight w_{intr}; labelled set D_s; reverse-trajectory set $\mathcal{X}_a$ from g_u.

1: Initialize main and target parameters: $\theta, \theta^{\text{trg}} \leftarrow \theta_0$.
2: **for** $k = 1, 2, \ldots$ **do**
3: Sample a mini-batch (x_i, y_i) from D_s and states x_t^τ from $\mathcal{X}_a$.
4: For each x_t^τ, draw one posterior sample $\tilde{x}_{t-1}^\tau \sim p(x_{t-1}|x_t^\tau)$ using g_u.
5: Compute the boundary loss $\mathcal{L}_{\text{bnd}}$ from Eq. equation 1.
6: Compute the anchored intermediate loss $\hat{\mathcal{L}}_{\text{intr}}^{\text{anc}}$ from Eq. equation 9, evaluating the labels at θ^{trg}.
7: Form the total loss $\mathcal{L} = \mathcal{L}_{\text{bnd}} + w_{\text{intr}} \hat{\mathcal{L}}_{\text{intr}}^{\text{anc}}$.
8: Update the main parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$.
9: Update the target parameters: $\theta^{\text{trg}} \leftarrow \alpha\, \theta^{\text{trg}} + (1 - \alpha)\, \theta$.

capturing the prior distribution of the data. It defines a reverse diffusion process over discrete time steps indexed by $t \in \{0, \ldots, T\}$, where $t = 0$ corresponds to the clean data space and $t = T$ to the fully noised space. A universe oracle f_u, trained on D_u in the original space, serves as the ground truth property evaluator. A reference oracle f_s, trained on D_s in the original space, approximates f_u under limited labelled data and is used to derive pseudo-labels.

To augment the training of f_θ beyond D_s, we generate a set of trajectories by reverse sampling from g_u. Each trajectory is indexed by $\tau \in \{1, \ldots, N_\tau\}$, and its intermediate states at all time steps are collected into the augmented set $\mathcal{X}_a = \{x_t^\tau \mid t \in \{0, \ldots, T\},\ \tau \in \{1, \ldots, N_\tau\}\}$. These trajectories are generated once prior to training and are fixed throughout. The reverse diffusion dynamics define a posterior distribution $p(x_{t-1} \mid x_t)$ over the less noisy state x_{t-1} given the current state x_t. The oracle f_θ takes an intermediate state x_t^τ as input; its time conditioning via the noise level is implicit and omitted from the notation for brevity. A full table of notation is provided in Appendix A.1.

3 Method

We train a property oracle f_θ using two sources of supervision: the small labelled reference dataset D_s, and the augmented trajectory states $\mathcal{X}_a$ generated by reverse sampling from g_u. The boundary loss supervises f_θ on D_s in the original space:

$$\mathcal{L}_{\text{bnd}} = \frac{1}{|D_s|} \sum_{(x_i, y_i) \in D_s} \mathcal{L}(y_i,\ f_\theta(x_i)). \tag{1}$$

The total training objective adds an intermediate loss on $\mathcal{X}_a$, weighted by w_{intr}:

$$\mathcal{L} = \mathcal{L}_{\text{bnd}} + w_{\text{intr}}\, \mathcal{L}_{\text{intr}}. \tag{2}$$

The key question is how to define $\mathcal{L}_{\text{intr}}$: a training signal for f_θ on the unlabelled intermediate states $\mathcal{X}_a$ (Figure 1). We first present our proposed approach, the Dynamics-Aware Predictor (DAP), then contrast it with an alternative that relies on pseudo-labels.

3.1 Dynamics-Aware Predictor (DAP)

DAP trains f_θ to respect the relationship between consecutive reverse diffusion step predictions. The reverse diffusion posterior $p(x_{t-1}|x_t)$ implies the following constraint on a well-calibrated oracle:

$$f_\theta(x_t) := \mathbb{E}\big[f_\theta(x_{t-1}) \mid x_t\big]. \tag{3}$$

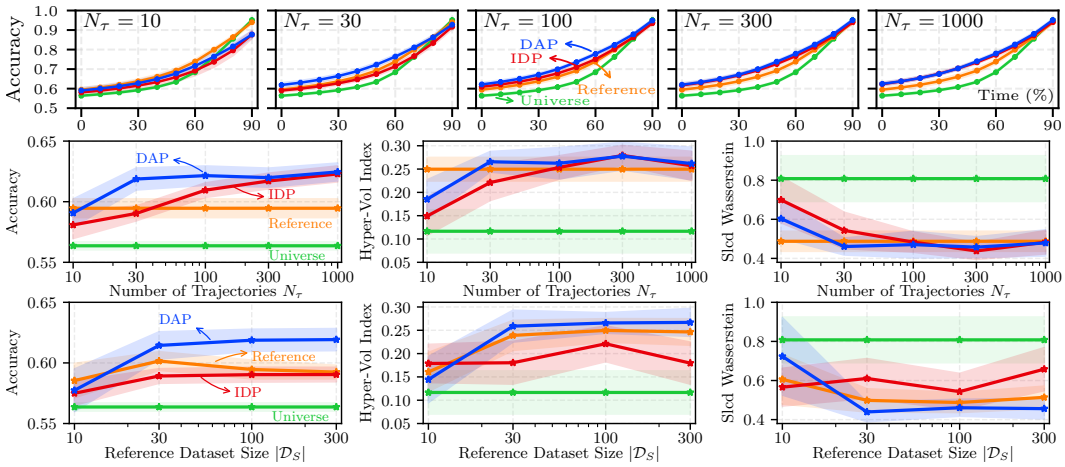


Figure 3: **DAP upper-bounds prediction and steering quality across data regimes on Half-Moon.** *Top*: DAP upper-bounds accuracy at most reverse time steps ($|D_s| = 100$); each value is averaged over $[t\text{–}100]\%$. *Middle*: accuracy averaged over time $[0\text{–}100]\%$; the gap between DAP and IDP widens with N_τ but narrows at very large trajectory counts, while DAP maintains the HVI upper bound ($|D_s| = 100$). *Bottom*: DAP upper-bounds accuracy and HVI across $|D_s|$ ($N_\tau = 30$).

This constraint serves as a learning signal that requires no labels for intermediate states; only the structure of the generative process matters. Enforcing it as a loss over the augmented trajectories defines the intermediate loss as an expectation over (τ, t):

$$\mathcal{L}_{\text{intr}} := \mathbb{E}_{\tau,t}\Big[\big\|f_\theta(x_t^\tau) - \mathbb{E}\big[f_\theta(x_{t-1}) \mid x_t^\tau\big]\big\|^2\Big]. \tag{4}$$

Reading Eq. equation 3 as a regression problem, $\mathbb{E}[f_\theta(x_{t-1}) \mid x_t]$ serves as the target label for $f_\theta(x_t)$, and substituting a single posterior sample $\tilde{x}_{t-1}^\tau \sim p(x_{t-1}|x_t^\tau)$ for this expectation gives the tractable estimator

$$\hat{\mathcal{L}}_{\text{intr}} := \mathbb{E}_{\tau,t}\Big[\big\|f_\theta(x_t^\tau) - f_\theta(\tilde{x}_{t-1}^\tau)\big\|^2\Big]. \tag{5}$$

In ordinary regression, such a substitution is harmless: θ-independent label noise averages out and leaves the optimum unchanged Robbins & Monro (1951); Kiefer & Wolfowitz (1952). Here, however, the label itself is a function of θ. Expanding the expectation over the posterior sample gives

$$\mathbb{E}_{\tilde{x}_{t-1}}\big[\hat{\mathcal{L}}_{\text{intr}}\big] = \mathcal{L}_{\text{intr}} + \mathbb{E}_{\tau,t}\big[\mathrm{Var}\big[f_\theta(x_{t-1}) \mid x_t^\tau\big]\big], \tag{6}$$

so the single-sample estimator is biased: its bias is a θ-dependent penalty on the prediction variance along the reverse chain that enters the gradient systematically and shifts the stationary point, pulling f_θ toward flat solutions that smooth the reverse dynamics away. A principled approach is to break the θ-coupling of the label via gradient detachment against an *anchored target*.

Anchored target. If the true solution θ^* were available, using $f_{\theta^*}(\tilde{x}_{t-1}^\tau)$ as the label would remove the θ-coupling entirely, and the resulting single-sample loss would share its stationary point with $\mathcal{L}_{\text{intr}}$:

Proposition 1. *Let θ^* satisfy Eq. equation 3 at every x_t, and define*

$$\hat{\mathcal{L}}^*_{intr}(\theta) := \mathbb{E}_{\tau,t}\big[\mathbb{E}_{\tilde{x}_{t-1}^\tau}\big[\|f_\theta(x_t^\tau) - f_{\theta^*}(\tilde{x}_{t-1}^\tau)\|^2\big]\big]. \tag{7}$$

Then $\nabla_\theta \hat{\mathcal{L}}^*_{intr}(\theta)\big|_{\theta=\theta^*} = \nabla_\theta \mathcal{L}_{intr}(\theta)\big|_{\theta=\theta^*} = 0$.

Proof. By assumption, $f_{\theta^*}(x_t) = \mathbb{E}[f_{\theta^*}(x_{t-1}) \mid x_t]$ at every x_t, so both the per-step residuals, and hence the gradients, of $\hat{\mathcal{L}}^*_{\text{intr}}$ and $\mathcal{L}_{\text{intr}}$ vanish at $\theta = \theta^*$. □

Replacing the θ-coupled label with a θ-independent target at θ^* eliminates the bias in Eq. equation 6 without perturbing the solution set. Since θ^* is the quantity we seek, the *anchored target* method

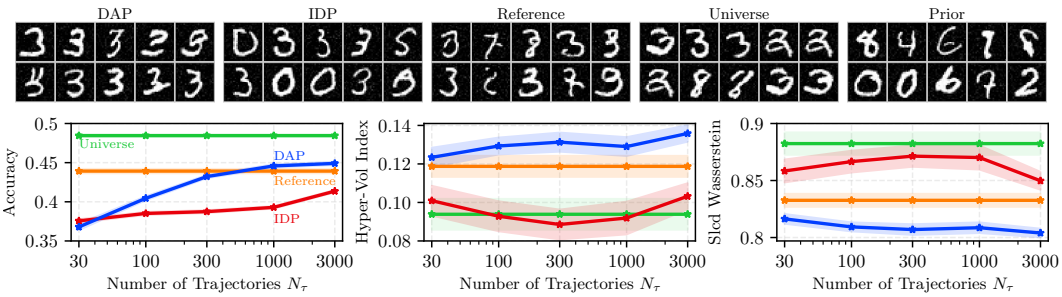


Figure 4: **DAP upper-bounds steering quality on MNIST and improves with more augmented trajectories; IDP underperforms even the reference oracle.** DAP generates diverse samples from the target class three, and maintains the HVI upper bound across all N_τ ($|D_s| = 1000$); the universe oracle (f_u) achieves high accuracy but yields inferior steering quality. Accuracy is averaged over the entire reverse time interval.

replaces this circular dependence with an iteratively refined estimate: θ^{trg} is maintained as an EMA of θ,

$$\theta^{\text{trg}} \leftarrow \alpha\, \theta^{\text{trg}} + (1-\alpha)\, \theta, \tag{8}$$

giving the anchored intermediate loss

$$\hat{\mathcal{L}}^{\text{anc}}_{\text{intr}} = \mathbb{E}_{\tau,t}\left[\left\| f_\theta(x^\tau_t) - f_{\theta^{\text{trg}}}(\tilde{x}^\tau_{t-1})\right\|^2\right]. \tag{9}$$

This severs the θ-coupling, so a single posterior sample per step suffices without inflating the loss by a variance term. The EMA keeps the target locally stationary, mirroring the iterative refinement of approximate dynamic programming Bertsekas & Tsitsiklis (1996); Sutton (1984); Mnih et al. (2015); Fujimoto et al. (2018), where Polyak-averaged targets are known to stabilize training Mnih et al. (2015); Fujimoto et al. (2018). The iterates converge to a *projected* fixed-point equation, which we state below.

What does anchored targeting optimize? Since the target in Eq. equation 9 moves with θ, there is no fixed global loss whose gradient drives the updates, making a direct theoretical analysis difficult. Under linear function approximation, however, this analysis becomes tractable: the iterates converge to a *projected* fixed point (the same "project after update" structure underlying approximate dynamic programming Tsitsiklis & Van Roy (1997)), with approximation error controlled solely by the function-class misspecification (see Appendix A.2).

Theorem 1. *Consider the linear function approximation class $f_\theta(x) = \phi(x)^\top \theta$, with feature matrix Φ, projection operator Π onto its span under the mixed training distribution, update operator $\mathcal{U}$ induced by the inter-step constraint paired with the boundary labels, and reverse-diffusion transition $\mathbf{P}^{\text{I}}$; full definitions are in Appendix A.2. Assume the sub-unit spectral radius condition $\sigma_{\mathbf{P}^{\text{I}}} < 1$, step sizes satisfying $\sum_k \eta_k = \infty$ and $\sum_k \eta_k^2 < \infty$, and a proper EMA rate α. Then the anchored-target iterates converge (in expectation) to a fixed point θ^*_{anc} of the projected update equation*

$$\Phi\, \theta^*_{anc} \;=\; \Pi \mathcal{U}\, \Phi\, \theta^*_{anc}. \tag{10}$$

Furthermore, letting f^* denote the true solution of the inter-step constraint coupled to the boundary labels, the mean-squared approximation error of θ^*_{anc} is bounded by the corresponding projection error of f^* onto the function-approximation class:

$$\mathbb{E}_x\left[(f_{\theta^*_{anc}}(x) - f^*(x))^2\right] \;\le\; \frac{1}{(1-\sigma_{\mathbf{P}^{\text{I}}})^2}\, \mathbb{E}_x\left[(\Pi f^*(x) - f^*(x))^2\right]. \tag{11}$$

Corollary 1 (Realizability). *If f^* lies in the function-approximation class, then $\Pi f^* = f^*$, both sides of Eq. equation 11 vanish, and θ^*_{anc} equals the true solution.*

Theorem 1 is proved under linear function approximation. In the non-linear case, training converges readily in practice; see Appendix A.4. The anchored-target method and its guarantees are grounded in dynamic programming theory Tsitsiklis & Van Roy (1997).

Appendix A.2 contains the full proof, including why the spectral condition reduces to a non-recirculation property of the reverse chain, and why the boundary loss is required to rule out the trivial constant solutions admitted by the inter-step constraint alone.

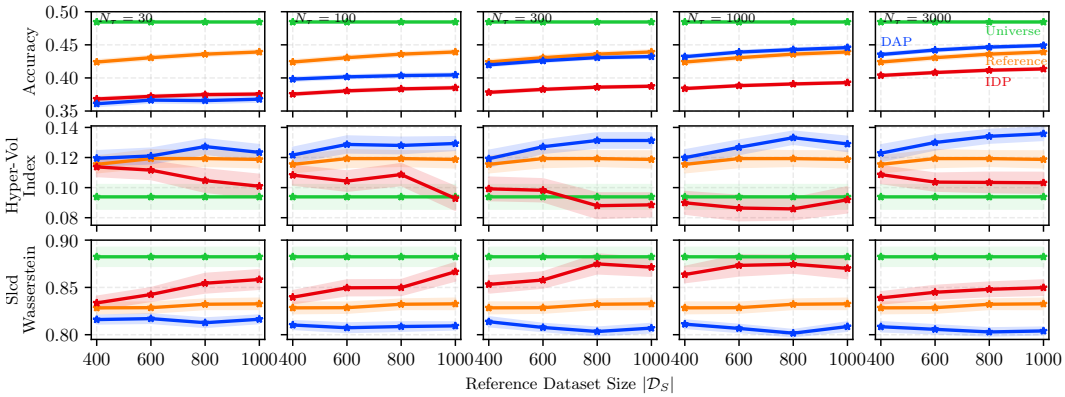


Figure 5: **DAP steering quality scales with reference dataset size on MNIST, with gains amplified by more trajectories.** *Top*: DAP accuracy averaged over time interval $[0\text{–}100]\%$ surpasses the reference oracle (f_s) as number of augmented trajectories (N_τ) increases and scales with reference dataset size ($|D_s|$); IDP underperforms f_s, suggesting that pseudo-labels are unreliable on the more diverse MNIST distribution. *Middle and bottom*: DAP upper-bounds steering quality; improvement with $|D_s|$ is more pronounced at larger N_τ.

Classification variant. The MSE formulation above applies to both regression and classification. For classification tasks, where f_θ outputs a class probability distribution $p_\theta(y|x_t)$, the anchored intermediate loss can equivalently be written as a cross-entropy against the target-network prediction as a soft label: $\hat{\mathcal{L}}_{\text{intr}}^{\text{anc}} = \text{CE}\big(p_{\theta^{\text{trg}}}(\cdot|\tilde{x}_{t-1}^\tau),\, p_\theta(\cdot|x_t^\tau)\big)$, and the same convergence argument applies. CE does not incur the variance bias from the θ-coupled label in the stochastic dynamics case; nonetheless, anchored targets remain beneficial, analogously to how semi-gradient methods outperform full-gradient alternatives even in deterministic dynamics Saleh & Jiang (2019); Fujimoto et al. (2022); Yin et al. (2022); Chen et al. (2021); Baird (1995) (see Appendix A.5).

3.2 Implicit-Dynamics Predictor (IDP)

An alternative approach to defining $\mathcal{L}_{\text{intr}}$ is to assign pseudo-labels to the trajectory states. The reference oracle f_s is evaluated at each trajectory endpoint x_0^τ to produce a pseudo-label $y_0^\tau = f_s(x_0^\tau)$. The IDP intermediate loss then directly maps each intermediate state to its trajectory pseudo-label:

$$\mathcal{L}_{\text{intr}}^{\text{IDP}} = \mathbb{E}_{\tau,t}[\mathcal{L}(y_0^\tau,\, f_\theta(x_t^\tau))]\,. \tag{12}$$

Since pseudo-labels are generated by f_s, the prediction quality of IDP is directly coupled to the generalizability of the reference oracle on the augmented trajectories.

4 Study Design

We conduct a systematic study comparing four oracle types: the universe oracle f_u and reference oracle f_s, which operate in the original data space, and the Implicit-Dynamics Predictor f_{IDP} and Dynamics-Aware Predictor f_{DAP}, which operate directly on the intermediate states (Figure 1). For the challenging molecular benchmark we also compare against the forward-noised reference oracle f_{fw} (Reference-fw), which is trained by forward-noising of the reference dataset samples Song et al. (2020) and operates directly on intermediate states. Since f_{fw} runs on infinite training sample budget, its performance was further studied under a relatively comparable training sample budget to DAP on the non-molecular benchmarks (see Appendix A.13).

Guidance. We use FK steering, which accommodates arbitrary black-box reward functions as well as predictor guidance, to evaluate each oracle as a steering signal. Since f_u and f_s operate only in the original data space, each intermediate state x_t is first mapped to the original space via the posterior mean $\mathbb{E}[x_0 \mid x_t]$, whose closed form is available for the VP-SDE Chung et al. (2022); the reward is then evaluated on this estimate. f_{IDP}, f_{DAP}, and f_{fw} operate directly on x_t, requiring no such mapping. Depending on whether the steering target is a class or a continuous property, the reward is defined as the predicted class probability or the property likelihood according to the oracle.

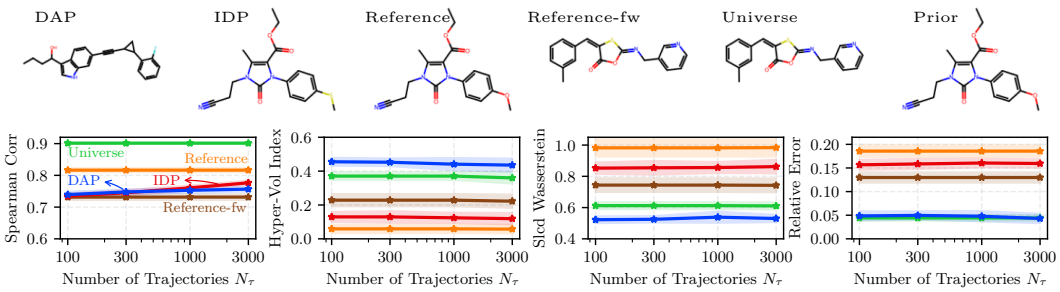


Figure 6: **DAP upper-bounds steering quality for molecule ring count in discrete flow guidance, consistently across all trajectory counts** ($|D_s| = 400$, targeting four molecule rings).

Prediction quality. We evaluate each oracle on a held-out set of sampling trajectories not used during training. At each time step t, we report classification accuracy and cross-entropy averaged over all held-out trajectories. As shown in Appendix A.2, the average cross-entropy at each t is minimised by the true conditional $p(y|x_t)$, which serves as the reference for oracle prediction quality. Spearman and pearson correlation coefficients are used for continuous properties.

Steering quality. We generate N trajectories with and without guidance and compare the two sets on three metrics. For class targets, *Delta mean reward* ($\Delta\bar{R} \in [0, 1]$) is the difference in mean reward between guided and prior samples, evaluated by the universe oracle f_u at the trajectory endpoints. For continuous targets, we instead report *Relative Error* (E): the absolute difference between the mean RDKit-computed property at the endpoints and the target value, normalised by the target. *SWD ratio* is the ratio of the Sliced Wasserstein Distance (SWD) between the generated set and the universe target set, divided by the SWD of the prior set to the same reference; a value of zero indicates perfect match to the target distribution, and a value above one indicates degradation relative to the prior. SWD is computed on embeddings derived from f_u (see Appendix A.8). To summarise reward and diversity simultaneously, we report the *Hypervolume Index* HVI $= \Delta\bar{R} \times (1 - \text{SWD ratio})$ Guerreiro et al. (2021), with $\Delta\bar{R}$ replaced by $1 - E$ for continuous targets; higher values are better (Appendix A.8).

5 Experiments

All oracles were trained, evaluated, and used for steering across combinations of reference dataset size $|D_s|$, number of augmented trajectories N_τ, and perturbation scheme: $|D_s|$ and N_τ characterise performance across data regimes, while three perturbation schemes applied to f_s probe IDP's sensitivity to reference oracle quality (Appendix A.9). Each configuration was repeated across random seeds (Tables 4 and 5), controlling dataset sampling, oracle training, and guidance; the universe oracle was trained once and shared. In total, ∼9000 configurations were evaluated across all benchmarks (Appendix A.7), with fixed DAP hyperparameters (w_{intr}, α) per benchmark (Appendix A.12).

DAP Produces Target-Confined Samples with Broad Coverage. In a low-data regime on Half-Moon ($|D_s|$=10, N_τ=30, Figure 2), DAP samples are tightly confined to the target class with broad coverage, whereas other oracles yield skewed coverage (f_u) or off-target samples (f_s, IDP), as confirmed quantitatively by HVI in subsequent sections.

5.1 DAP Upper-Bounds Prediction and Steering Quality on Half-Moon

DAP consistently upper-bounds prediction and steering quality across a wide range of data regimes, achieving this without pseudo-label supervision and remaining robust. Figure 3 (top) shows averaged accuracy over $[t\text{–}100]\%$ as a function of time t for each N_τ. DAP achieves the highest prediction accuracy at most time percentiles. The accuracy of f_u and f_s is unaffected by N_τ since augmented trajectories play no role in their training and both map intermediate states via the posterior mean, while DAP and IDP operate directly on x_t. Figures 24 and 26 confirm that this holds across most $|D_s|$ and N_τ, with DAP achieving the best accuracy and cross entropy.

Figure 3 (middle). Accuracy averaged over the full time interval $[0\text{–}100]\%$, SWD ratio, and HVI are shown as a function of N_τ. The accuracy of DAP surpasses that of the reference oracle as the number

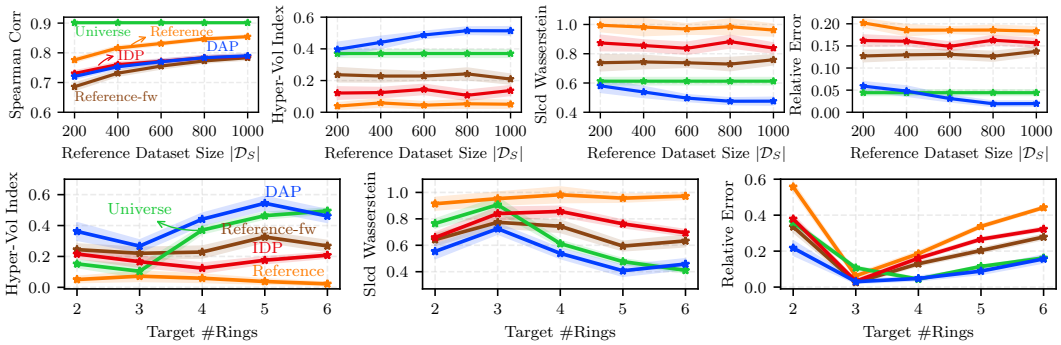


Figure 7: **DAP prediction and steering quality scales with reference dataset size and generalizes across targets in discrete flow guidance.** *Top*: Larger $|D_s|$ improves steering for molecule ring count of four ($N_\tau = 1000$). Spearman is averaged over time interval $[0, 100]\%$. *Bottom*: DAP upper-bounds HVI and lower-bounds SWD across most target ring counts ($N_\tau = 1000$, $|D_s| = 400$).

of trajectories N_τ increases. Furthermore, the accuracy gap between DAP and IDP widens as N_τ increases; at very large N_τ the gap narrows, yet DAP maintains the upper bound on HVI. Figure 22 confirms that DAP achieves the best accuracy and HVI across all N_τ values in most cases.

Figure 3 (bottom). The same metrics as the middle panel are shown as a function of $|D_s|$. DAP achieves the best accuracy and HVI. Figure 27 confirms this pattern holds across most $|D_s|$ and N_τ combinations (see Appendix A.11).

Since DAP does not rely on pseudo-labels, it is unaffected by reference oracle degradation, whereas IDP degrades under three perturbation schemes of increasing aggressiveness (Appendix A.9).

5.2 DAP Steering Quality Improves with Augmented Trajectories and Scales with Reference Dataset Size on MNIST

On MNIST, DAP upper-bounds steering quality across all N_τ and its prediction accuracy improves with N_τ (Figures 4 and 25), while IDP underperforms even f_s, suggesting pseudo-labels are unreliable on the more diverse MNIST distribution. Notably, f_u achieves high accuracy yet inferior steering quality. DAP achieves the best steering quality as a function of N_τ, regardless of $|D_s|$ (Figure 23).

Steering quality also scales with $|D_s|$ (Figure 5). For all trajectory counts, DAP upper-bounds steering quality, and the improvement with $|D_s|$ is more pronounced at larger N_τ. Similarly, prediction accuracy improves with $|D_s|$, with gains amplified at larger N_τ (Figure 28). DAP generates more diverse samples from target class three than other oracles (Figure 4 (top) and Figure 9).

5.3 DAP Upper-Bounds Steering Quality in Discrete Flow Guidance on Molecules

For steering a discrete flow model towards a target number of rings, DAP consistently outperforms all non-universe oracles regardless of data regime, maintaining superior steering quality across varying numbers of augmented trajectories (Figures 6 and 10) and scaling with reference dataset size (Figure 7, top, and Figure 29).

Across varying target ring counts, DAP upper-bounds the steering quality of other oracles (Figure 7, bottom). In some settings, f_u outperforms DAP for one target value distant from the training distribution median (Figure 11). The forward-noised oracle ranks second or third in steering performance and almost always outperforms IDP (Figures 11 and 12).

Similar observations hold for lipophilicity steering: DAP mostly upper-bounds steering quality across trajectory counts, reference dataset sizes, and target property values (Figures 13 and 31). Compared to the ring-count steering, the forward-noised oracle is the second-best oracle in steering performance more frequently than the universe oracle (Figures 14 and 15).

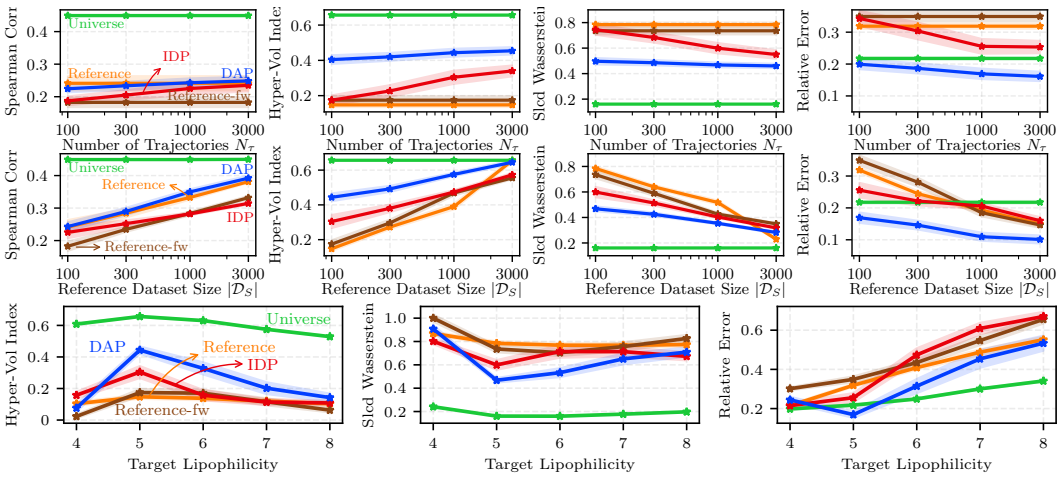


Figure 8: **DAP upper-bounds steering quality in predictor guidance on the latent space of molecules.** *Top*: DAP maintains the best steering quality across augmented trajectory counts ($|D_s| = 100$). *Middle*: DAP steering quality scales with reference dataset size ($N_\tau = 1000$, target lipophilicity $= 5$). *Bottom*: DAP maintains the upper bound of steering quality among non-universe oracles for most target lipophilicity values.

DAP's prediction performance improves as N_τ increases, with more pronounced gains at smaller $|D_s|$ (Figures 10 and 13). DAP matches or exceeds the prediction performance of the forward-noised oracle. Reference oracle f_s achieves better prediction but worse steering than DAP.

5.4 DAP Upper-Bounds Steering Quality in Classifier Guidance on Molecular Latent Space

For steering towards a target lipophilicity value on the latent space of VAE-encoded molecules, DAP upper-bounds all non-universe oracles (Figure 8, top). DAP's steering performance improves with increasing N_τ, an effect more pronounced at smaller $|D_s|$ (Figure 16). DAP's prediction and steering quality also scale with $|D_s|$ (Figure 8, middle, and Figure 33).

Across varying target lipophilicity values, DAP mostly upper-bounds non-universe oracles (Figure 8, bottom). For target values farther from the training distribution median, the gap between DAP and the forward-noised oracle narrows, with DAP remaining superior. With increased $|D_s|$, f_s achieves better steering at these extreme target values (Figures 17 and 18).

DAP's performance is stable across a wide range of EMA window sizes (25–100) and robust across three orders of magnitude in w_{intr}, with the latter having a more pronounced effect on steering quality at smaller $|D_s|$ (Appendix A.12).

6 Conclusion

We studied property oracle training for steering continuous diffusion and discrete flow models in the low-data regime, proposing DAP: an oracle trained on unlabelled reverse trajectories under an inter-step dynamics constraint, with an anchored-target optimization scheme supported by theoretical analysis. DAP achieves the best steering quality across our benchmark on classification and regression tasks in both discrete and continuous design spaces, showing that prediction accuracy alone does not determine steering performance.

Enforcing reverse dynamics is most valuable in the label-scarce regime common to scientific applications, where direct supervision of intermediate states is unreliable; DAP provides this without pseudo-label supervision. Our experiments established DAP's advantage with a single fixed hyperparameter configuration across all settings (Appendix A.11); further gains can be expected from per-application tuning.

REFERENCES

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pp. 30–37. Elsevier, 1995.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 843–852, 2023.

Dimitri P Bertsekas. A counterexample to temporal differences learning. *Neural computation*, 7(2): 270–279, 1995.

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5): 291–294, 1997.

Justin Boyan and Andrew W Moore. Generalization in reinforcement learning: Safely approximating the value function. *Advances in neural information processing systems*, pp. 369–376, 1995.

Gabriel Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided diffusion for bayesian linear inverse problems. *arXiv preprint arXiv:2308.07983*, 2023.

Yutian Chen, Liyuan Xu, Caglar Gulcehre, Tom Le Paine, Arthur Gretton, Nando de Freitas, and Arnaud Doucet. On instrumental variable regression for deep offline policy evaluation. *arXiv preprint arXiv:2105.10148*, 2021.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Peter Dayan. The convergence of TD(λ) for general λ. *Machine learning*, 8(3-4):341–362, 1992.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M Bronstein, et al. Scaling atomistic protein binder design with generative pretraining and test-time compute. *arXiv preprint arXiv:2603.27950*, 2026.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024.

Peter Eckmann, Kunyang Sun, Bo Zhao, Mudong Feng, Michael K Gilson, and Rose Yu. Limo: Latent inceptionism for targeted molecule generation. *Proceedings of machine learning research*, 162:5777, 2022.

Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. Why should I trust you, Bellman? the Bellman error is a poor replacement for value error. *arXiv preprint arXiv:2201.12417*, 2022.

Xuefeng Gao, Jiale Zha, and Xun Yu Zhou. Reward-directed score-based diffusion models via q-learning. *Journal of Machine Learning Research*, 26(302):1–46, 2025.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1):163–223, 2003.

Geoffrey J Gordon. Stable function approximation in dynamic programming. In *Proceedings of the twelfth international conference on machine learning*, pp. 261–268, 1995.

Andreia P Guerreiro, Carlos M Fonseca, and Luís Paquete. The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys (CSUR)*, 54(6):1–42, 2021.

Erik Hartman, Jonas Wallin, Johan Malmström, and Jimmy Olsson. Controllable protein design through feynman-kac steering. *arXiv preprint arXiv:2511.09216*, 2025.

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

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

Clemens Isert, Kenneth Atz, José Jiménez-Luna, and Gisbert Schneider. Qmugs, quantum mechanical properties of drug-like molecules. *Scientific Data*, 9(1):273, 2022.

Yazid Janati, Badr Moufad, Alain Durmus, Eric Moulines, and Jimmy Olsson. Divide-and-conquer posterior sampling for denoising diffusion priors. *Advances in Neural Information Processing Systems*, 37:97408–97444, 2024.

Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics*, pp. 531–539, 2006.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 22825–22855. PMLR, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Marie Oestreich, Erinc Merdivan, Michael Lee, Joachim L Schultze, Marie Piraud, and Matthias Becker. Drugdiff: small molecule diffusion model with flexible guidance towards molecular properties. *Journal of cheminformatics*, 17(1):23, 2025.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Ehsan Saleh and Nan Jiang. Deterministic Bellman residual minimization. In *Proceedings of Optimization Foundations for Reinforcement Learning Workshop at NeurIPS*, 2019.

Jian Shen, Shengmin Zhou, and Xing Che. Fksfold: Improving alphafold3-type predictions of molecular glue–induced ternary complexes with feynman–kac–steered diffusion. *bioRxiv*, pp. 2025–05, 2025.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Richard Stuart Sutton. *Temporal credit assignment in reinforcement learning*. University of Massachusetts Amherst, 1984.

Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1):59–94, 1996.

John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5), 1997.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Shuyu Yin, Tao Luo, Peilin Liu, and Zhi-Qin John Xu. An experimental comparison between temporal difference and residual gradient with neural network approximation. *arXiv preprint arXiv:2205.12770*, 2022.

Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Minshuo Chen, and Mengdi Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. *Advances in Neural Information Processing Systems*, 36:60599–60635, 2023.

Zheng Zhao, Ziwei Luo, Jens Sjölund, and Thomas Schön. Conditional sampling within generative diffusion models. *Philosophical Transactions A*, 383(2299):20240329, 2025.

A Appendix

A.1 Notation

Table 1: Summary of Notation.

Symbol	Description
D_u	Universe dataset (all samples with labels)
D_s	Reference dataset (small labelled subset, $D_s \subset D_u$)
D_a	Augmented dataset: intermediate states with pseudo-labels $\{(x_t^\tau, y_0^\tau)\}$
$\mathcal{X}_a$	Augmented set of intermediate states (unlabelled): $\{x_t^\tau\}$
$\mathcal{Y}_a$	Pseudo-labels for augmented trajectories: $\{y_0^\tau\}$
g_u	Pretrained generative model trained on D_u
f_u	Universe oracle trained on D_u
f_s	Reference oracle trained on D_s
f_θ	Oracle being trained (DAP or IDP)
$\tau \in \{1, \dots, N_\tau\}$	Trajectory index
$t \in \{0, \dots, T\}$	Time step ($t = 0$: clean space, $t = T$: fully noised)
x_t^τ	Intermediate state at time t on trajectory τ
$y_0^\tau = f_s(x_0^\tau)$	Pseudo-label for trajectory τ
$p(x_{t-1}\|x_t)$	Reverse diffusion posterior
$\tilde{x}_{t-1}^\tau$	Single sample drawn from $p(x_{t-1}\|x_t^\tau)$ at training time
θ^{trg}	Moving average (Polyak) target network parameters
α	Moving average update rate
w_{intr}	Relative weight of the intermediate loss
$\Delta\bar{R}$	Delta mean reward (guided minus prior, evaluated by f_u)
SWD ratio	SWD(guided, target) / SWD(prior, target)
HVI	Hypervolume Index: $\Delta\bar{R} \times (1 - \text{SWD ratio})$

A.2 Anchored-Target Theory: Definitions, Theorem, and Proofs

Since the target in Eq. equation 9 moves with θ, there is no fixed loss whose gradient exactly drives the updates. A clean characterization is nonetheless available under linear function approximation: the iterates converge to the fixed point of a *projected* update operator: first, propagate predictions one step along the reverse chain using the inter-step constraint; second, project the result back onto the function-approximation class under the training distribution. The same "project after update" fixed-point structure underlies the classical analysis of approximate dynamic programming with function approximation Tsitsiklis & Van Roy (1997). When the function class realizes the true solution, the projection is an identity and the anchored target recovers it exactly; when it does not, the projected fixed point remains a well-defined limit and its distance from the true solution is controlled by the function-class misspecification alone.

Here, we restate and prove Theorem 1, together with its supporting definitions, assumptions, and a proof following the analysis of approximate dynamic programming under linear function approximation Tsitsiklis & Van Roy (1997); Bertsekas & Tsitsiklis (1996); Tsitsiklis & Van Roy (1996).

Table 2: Overview of benchmarks, generative models, and steering setups. *Predictor guidance* refers to gradient-based predictor guidance throughout. On QMugs under discrete flow guidance, steering was carried out for two different properties (number of rings and lipophilicity), giving three molecular steering tasks in total.

Dataset	Generative model	Steering task	Steering method
Half-Moon	Continuous diffusion	Classification	FK steering Singhal et al. (2025)
MNIST	Continuous diffusion	Classification	FK steering
QMugs	Discrete flow	Regression	Predictor guidance Nisonoff et al. (2024)
QMugs	Continuous diffusion	Regression	Predictor guidance Song et al. (2020)

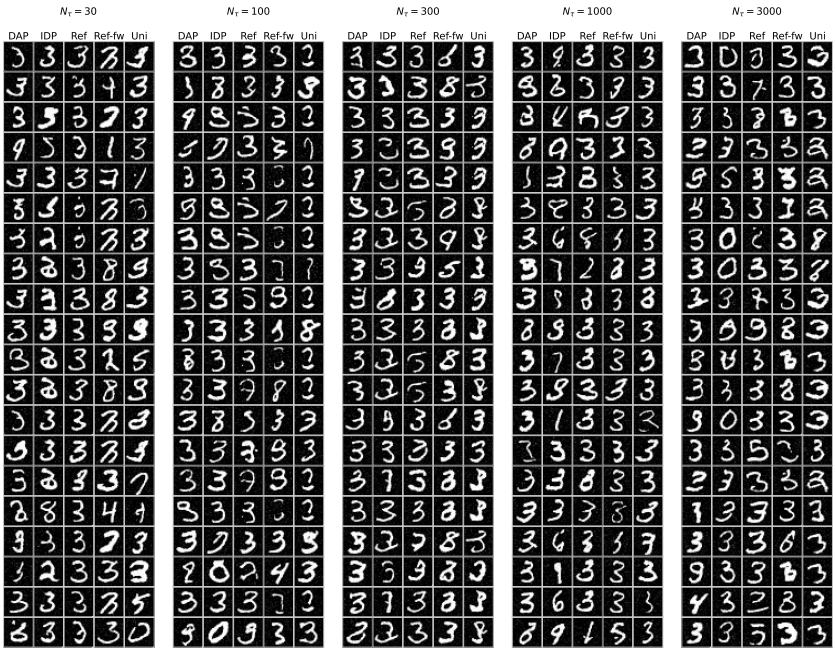


Figure 9: MNIST samples generated by guidance using different oracles at reference dataset size $|D_s| = 1000$. Each column corresponds to a different number of augmented trajectories used for training IDP and DAP; 20 guided samples per oracle are shown. DAP produces more diverse samples from the target class, confirming that its steering quality upper-bounds the other oracles; other oracles generate more off-target samples or show less diversity within the target class.

A.2.1 Setting, Definitions, and Assumptions

State-space discretization. For the theoretical analysis we discretize the continuous input domain $\mathcal{X}$ into an abstract space of S bins, $\mathcal{X} \cong \{x_1, x_2, \ldots, x_S\}$, following the abstraction framework for large state spaces used in the analysis of Markov processes Li et al. (2006); Givan et al. (2003). We take $S \to \infty$ so no expressive power is lost; the abstraction merely lets us write every quantity of interest in matrix form rather than as an integral, and has no bearing on the algorithm itself, which operates directly in the original continuous domain.

Linear function approximation and feature maps. Throughout this appendix we work with the linear function class

$$f_\theta(x) = \phi(x)^\top \theta, \qquad \phi(x) \in \mathbb{R}^d, \qquad \theta \in \mathbb{R}^d, \tag{13}$$

and we let

$$\Phi := [\phi(x_1), \ldots, \phi(x_S)]^\top \in \mathbb{R}^{S \times d} \tag{14}$$

denote the stacked feature matrix.

Table 3: Oracle types, training data, operating space, and input during guidance.

Oracle	Symbol	Training data	Operating space	Input
Universe oracle	f_u	D_u	Original ($t = 0$)	$\mathbb{E}[x_0 \mid x_t]$
Reference oracle	f_s	D_s	Original ($t = 0$)	$\mathbb{E}[x_0 \mid x_t]$
Implicit-Dynamics Predictor	f_{IDP}	$D_s \cup D_a$	Intermediate ($0 \le t \le T$)	x_t
Dynamics-Aware Predictor	f_{DAP}	$D_s, \mathcal{X}_a$	Intermediate ($0 \le t \le T$)	x_t
Forward-Noised Oracle	f_{fw}	D_s, forward-noised	Intermediate ($0 \le t \le T$)	x_t

Table 4: Experimental grid for Half-Moon and MNIST benchmarks.

Variable	Half-Moon	MNIST
Reference dataset size ($\lvert D_s \rvert$)	10, 30, 100, 300	400, 600, 800, 1000
Number of augmented trajectories (N_τ)	10, 30, 100, 300, 1000	30, 100, 300, 1000, 3000
Perturbation schemes	uniform, shuffle, adversary	—
Perturbation extents	0, 0.2, 0.4, 0.6, 0.8	—
Random seeds (full pipeline)	20	100
Total configurations	5200	2000
GPU hours	1300	1000
GPU memory per configuration	<20 GB	∼20 GB

Sampling distributions and their matrices. The training objective Eq. equation 2 is driven by two sources of supervision, each with its own sampling distribution on $\mathcal{X}$:

- a *boundary* distribution P_{bnd}, supported on D_s (the clean, labelled points at $t = 0$), with associated labels $y : \mathcal{X} \to \mathbb{R}$ available exactly on $\operatorname{supp}(P_{\text{bnd}})$;
- an *intermediate* distribution P_{intr}, supported on the reverse-trajectory states in $\mathcal{X}_a$ at all $t \geq 1$, paired with the one-step reverse-diffusion posterior $p(x_{t-1}|x_t)$.

For the remainder, the state space $\mathcal{X}$ is understood as the combined training support $\operatorname{supp}(P_{\text{bnd}}) \cup \operatorname{supp}(P_{\text{intr}})$, and P_{intr} is taken to include trajectory endpoints at $t = 0$ (so that one-step push-forwards from $t = 1$ land in $\operatorname{supp}(P_{\text{intr}})$ by construction). We emphasise that P_{bnd} and P_{intr} are *distributions* on $\mathcal{X}$, not the finite sample sets D_s and $\mathcal{X}_a$ themselves: the entire analysis is carried out at the level of measures, never at the level of individual trajectories and their endpoints. In particular, nothing below requires that a sampled reverse trajectory terminate at, or ever visit, a point of D_s. At $t = 0$ the two measures describe the same law: D_s is a labelled subset of D_u, and g_u approximates the distribution of D_u, so the trajectory endpoints and the labelled points are drawn from a common distribution and differ only in which trajectories produced them. The structure is that of a boundary-value problem: labels constrain the solution on $\operatorname{supp}(P_{\text{bnd}})$, the inter-step constraint constrains it on $\operatorname{supp}(P_{\text{intr}})$, and a single function class must satisfy both. We write $D_{\text{bnd}} := \operatorname{diag}(P_{\text{bnd}})$ and $D_P := \operatorname{diag}(P_{\text{intr}})$ for the associated diagonal matrices, and $Y_{\text{bnd}} \in \mathbb{R}^S$ for the label vector with $(Y_{\text{bnd}})_x = y(x)$ on $\operatorname{supp}(P_{\text{bnd}})$ and zero elsewhere. The one-step reverse transition is encoded by

$$\mathbf{P}^{\downarrow} \in \mathbb{R}^{S \times S}, \qquad (\mathbf{P}^{\downarrow})_{ij} := p(x' = x_j \mid x = x_i), \tag{15}$$

so that $(\mathbf{P}^{\downarrow} H)(x) = \mathbb{E}[H(x') \mid x]$, and boundary points are absorbing under this operator (the chain terminates upon hitting $\operatorname{supp}(P_{\text{bnd}})$). This is a convention about how $\mathcal{U}$ is defined piecewise, not a requirement that the reverse chain reach the labelled points. The *mixed training measure* is

$$Q := D_{\text{bnd}} + w_{\text{intr}}\, D_P. \tag{16}$$

Table 5: Experimental grid for the QMugs benchmark. The QMugs DFM grid is shared across both guided properties (lipophilicity and number of rings).

Variable	QMugs (DFM)	QMugs (latent diffusion)
Reference dataset size ($\lvert D_s \rvert$)	200, 400, 600, 800, 1000	100, 300, 1000, 3000
Augmented trajectories (N_τ)	100, 300, 1000, 3000	100, 300, 1000, 3000
Target property value (y_{target})	2, 3, 4, 5, 6	4, 5, 6, 7, 8
Random seeds (full pipeline)	30	30
Total configurations	600	960
GPU hours	1200	800
GPU memory per configuration	∼20 GB	∼20 GB

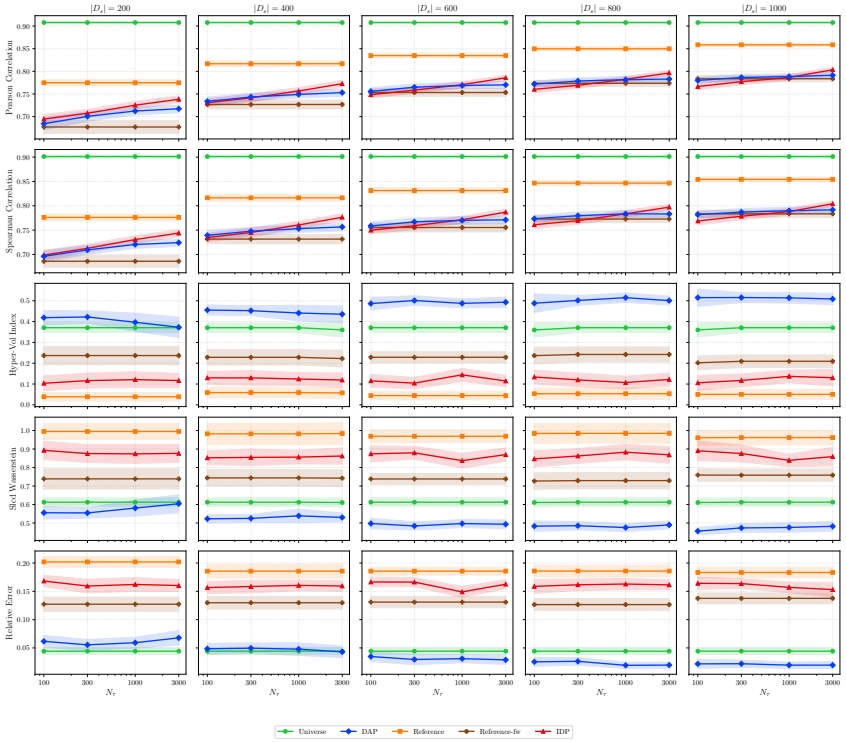


Figure 10: Prediction and steering quality as a function of the number of augmented trajectories for the QMugs benchmark under discrete flow guidance, targeting four molecule rings. Prediction quality is measured by Spearman and Pearson correlation coefficients averaged over the time interval $[0\text{–}100]\%$. Each column corresponds to a different reference dataset size.

Update and projection operators. Combining the two supervision sources, we define the update operator $\mathcal{U} : \mathbb{R}^S \to \mathbb{R}^S$ that encodes the desired fixed point:

$$(\mathcal{U}\, H)(x) = \begin{cases} y(x), & x \in \mathrm{supp}(P_{\mathrm{bnd}}), \\ (\mathbf{P}^{\mid}\, H)(x), & x \in \mathrm{supp}(P_{\mathrm{intr}}). \end{cases} \tag{17}$$

In words: on the labelled boundary, $\mathcal{U}$ returns the

clean-space label; on intermediate states, it returns one step of reverse-diffusion propagation applied to the current function. The *projection operator* Π is the weighted L^2 projection onto $\mathrm{span}(\Phi)$ under Q:

$$\Pi\, H := \Phi\, (\Phi^\top Q\, \Phi)^{-1}\, \Phi^\top Q\, H, \qquad \|\mathbf{H}\|_Q^2 := \mathbf{H}^\top Q\, \mathbf{H}. \tag{18}$$

We denote by $\sigma_{\mathbf{P}^{\mid}}$ the smallest constant satisfying the pointwise domination of the weighted push-forward by Q:

$$\sigma_{\mathbf{P}^{\mid}}^2 \;:=\; \max_{x' \in \mathcal{X}} \frac{w_{\mathrm{intr}}\, (P_{\mathrm{intr}}\, \mathbf{P}^{\mid})(x')}{Q(x')}, \tag{19}$$

which is finite by (A2). The mixed measure Q is the natural norm for bounding $\mathbf{P}^{\mid}$: the D_P-weighted operator norm is not finite in general, since $\mathbf{P}^{\mid}$ feeds boundary values into intermediate entries while $\|\cdot\|_{D_P}$ puts zero weight on $\mathrm{supp}(P_{\mathrm{bnd}})$.

Assumptions. We assume:

(A1) **Bounded, full-rank features:** $\|\phi(x)\|_2 \le 1$ for all $x \in \mathcal{X}$; $\mathrm{rank}(\Phi) = d$.

(A2) **Full-support sampling:** $P_{\mathrm{bnd}}(x) + w_{\mathrm{intr}} P_{\mathrm{intr}}(x) > 0$ for all $x \in \mathcal{X}$, so Q is positive definite.

(A3) **Strict contraction property:** $\sigma_{\mathbf{P}^{\mid}} < 1$, with $\sigma_{\mathbf{P}^{\mid}}$ as defined in Eq. equation 19.

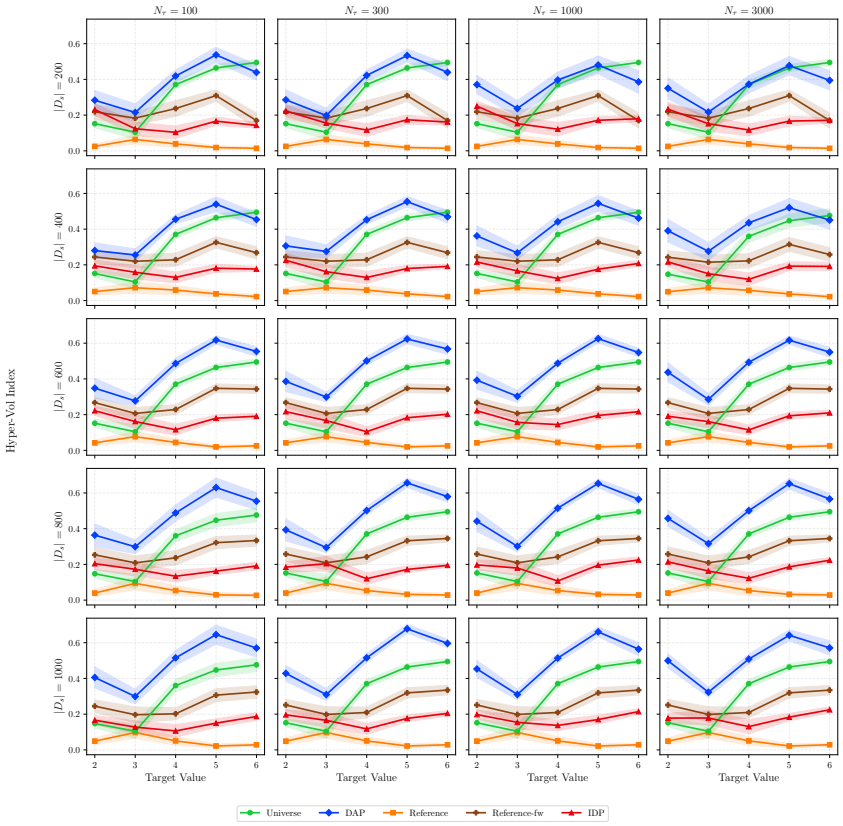


Figure 11: HVI as a function of the target number of rings for the QMugs benchmark under discrete flow guidance. Each column corresponds to a different number of augmented trajectories and each row to a different reference dataset size.

(A4) **Robbins–Monro step sizes:** $\sum_k \eta_k = \infty$, $\sum_k \eta_k^2 < \infty$.

(A5) **Slow target.** The EMA rate α is set on a slower timescale than η_k (Polyak averaging Fujimoto et al. (2018); Borkar (1997)), so θ^{trg} tracks θ in expectation at the fixed point.

Assumption (A3) is the non-trivial one; its role in the DAP setting is discussed in Section A.2.3 below.

Fixed-point reformulation. With these definitions in hand, the solution target of DAP can be re-stated as the fixed point of the combined update operator:

$$\Phi\,\theta^* = \mathcal{U}\,\Phi\,\theta^* \quad \Longleftrightarrow \quad \begin{cases} f_{\theta^*}(x) = y(x), & x \in \operatorname{supp}(P_{\text{bnd}}), \\ f_{\theta^*}(x_t) = \mathbb{E}[f_{\theta^*}(x_{t-1}) \mid x_t], & x_t \in \operatorname{supp}(P_{\text{intr}}). \end{cases} \tag{20}$$

The anchored-target method solves a *projected* variant of this equation; this is the content of Theorem 1.

A.2.2 Restatement and Proof of Theorem 1

Theorem 1 (Convergence and approximation error of the anchored target; restated)**.** *Under assumptions (A1)–(A5), the anchored-target iterates converge (in expectation) to a fixed point θ^*_{anc} of the projected update equation*

$$\Phi\,\theta^*_{anc} = \Pi\mathcal{U}\,\Phi\,\theta^*_{anc}. \tag{21}$$

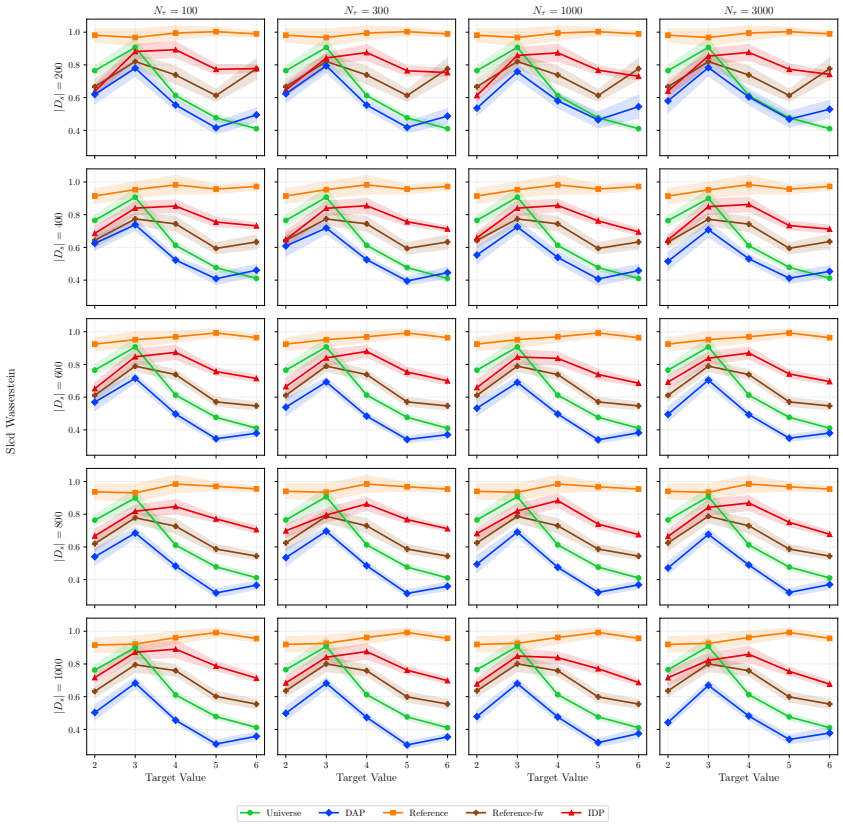


Figure 12: SWD ratio as a function of the target number of rings for the QMugs benchmark under discrete flow guidance. Each column corresponds to a different number of augmented trajectories and each row to a different reference dataset size.

Letting $f^* \in \mathbb{R}^S$ be the true solution of Eq. equation 20, the mean-squared approximation error of θ^*_{anc} in the mixed training norm is bounded by the projection error of f^*:

$$\|\Phi\,\theta^*_{anc} - f^*\|^2_Q \;\le\; \frac{1}{(1-\sigma_{\mathbf{P}^|})^2}\,\|\Pi f^* - f^*\|^2_Q. \tag{22}$$

Corollary 2 (Realizability; Cor. 1 restated)**.** *If f^* lies in the span of Φ, then $\Pi f^* = f^*$ and both sides of Eq. equation 22 vanish, so θ^*_{anc} equals the true solution.*

Proof. The proof proceeds in two stages. Stage 1 shows that the anchored-target updates are an instance of stochastic approximation whose expected fixed point satisfies Eq. equation 21.

Stage 2 upper-bounds the approximation error of that fixed point via a triangle-inequality chain, using the non-expansivity of Π and the contraction property of $\mathbf{P}^|$ on $\|\cdot\|_Q$.

Stage 1 (stochastic approximation to the projected fixed point). Under the two-timescale scaling in (A5), θ evolves on the fast timescale while θ^{trg} tracks it on a slower one, so at any fixed point of the coupled iteration we have $\theta^{\text{trg}} = \theta$ in expectation Borkar (1997). Substituting this fixed-point identity into Eq. equation 9 makes the anchored intermediate gradient equal (in expectation) to the gradient of the squared residual on P_{intr}, with the target branch treated as a constant. The combined update per training step is therefore an instance of linear stochastic recursion:

$$\theta_{k+1} \leftarrow \theta_k - \eta_k\,(A_k\,\theta_k - b_k), \tag{23}$$

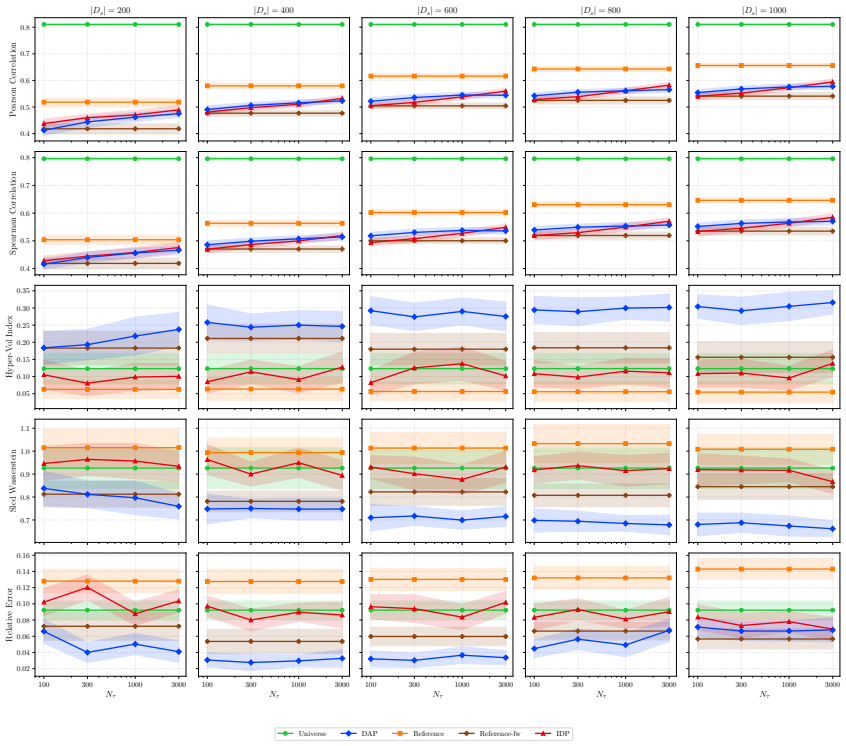


Figure 13: Prediction and steering quality as a function of the number of augmented trajectories for the QMugs benchmark under discrete flow guidance, targeting a lipophilicity of four. Prediction quality is measured by Spearman and Pearson correlation coefficients averaged over the time interval $[0\text{–}100]\%$. Each column corresponds to a different reference dataset size.

where, for one sampled boundary point x_{bnd} with label $y(x_{\text{bnd}})$ and one sampled intermediate pair $(x_t^\tau, \tilde{x}_{t-1}^\tau)$,

$$A_k = \phi(x_{\text{bnd}})\phi(x_{\text{bnd}})^\top \; + \; w_{\text{intr}}\, \phi(x_t^\tau)\big(\phi(x_t^\tau) - \phi(\tilde{x}_{t-1}^\tau)\big)^\top, \tag{24}$$

$$b_k = \phi(x_{\text{bnd}})\, y(x_{\text{bnd}}). \tag{25}$$

Taking expectations over the sampling distributions gives

$$\mathbf{A} := \mathbb{E}[A_k] = \Phi^\top D_{\text{bnd}}\, \Phi \; + \; w_{\text{intr}}\, \Phi^\top D_P\, (I - \mathbf{P}^{\text{|}})\, \Phi, \tag{26}$$

$$\mathbf{b} := \mathbb{E}[b_k] = \Phi^\top D_{\text{bnd}}\, Y_{\text{bnd}}. \tag{27}$$

Under (A1)–(A4), the Robbins–Monro theorem for linear stochastic approximation Robbins & Monro (1951); Kiefer & Wolfowitz (1952); Tsitsiklis & Van Roy (1997) guarantees that the iterates converge (in expectation) to the solution of the deterministic linear system $\mathbf{A}\,\theta^*_{\text{anc}} = \mathbf{b}$. It remains to verify that this solution satisfies Eq. equation 21. Writing $\mathbf{A}\,\theta^*_{\text{anc}} = \mathbf{b}$ in full,

$$\Phi^\top Q\, \Phi\, \theta^*_{\text{anc}} = \Phi^\top D_{\text{bnd}}\, Y_{\text{bnd}} \; + \; w_{\text{intr}}\, \Phi^\top D_P\, \mathbf{P}^{\text{|}}\, \Phi\, \theta^*_{\text{anc}}. \tag{28}$$

By the definition of $\mathcal{U}$ (Eq. equation 17), $Q\,\mathcal{U}\,\Phi\,\theta = D_{\text{bnd}}\, Y_{\text{bnd}} + w_{\text{intr}}\, D_P\, \mathbf{P}^{\text{|}}\, \Phi\, \theta$, so Eq. equation 28 reads $\Phi^\top Q\, \Phi\, \theta^*_{\text{anc}} = \Phi^\top Q\, \mathcal{U}\, \Phi\, \theta^*_{\text{anc}}$. Pre-multiplying by $(\Phi^\top Q\, \Phi)^{-1}\Phi$ and invoking Eq. equation 18 yields $\Phi\, \theta^*_{\text{anc}} = \Pi\mathcal{U}\, \Phi\, \theta^*_{\text{anc}}$, as claimed.

Stage 2 (triangle-inequality chain). We now bound the approximation error $\|\Phi\theta^*_{\text{anc}} - f^*\|_Q$, where f^* is the true solution satisfying $f^* = \mathcal{U} f^*$ (Eq. equation 20). Using $\Phi\theta^*_{\text{anc}} = \Pi\mathcal{U}\, \Phi\theta^*_{\text{anc}}$ and Πf^* as

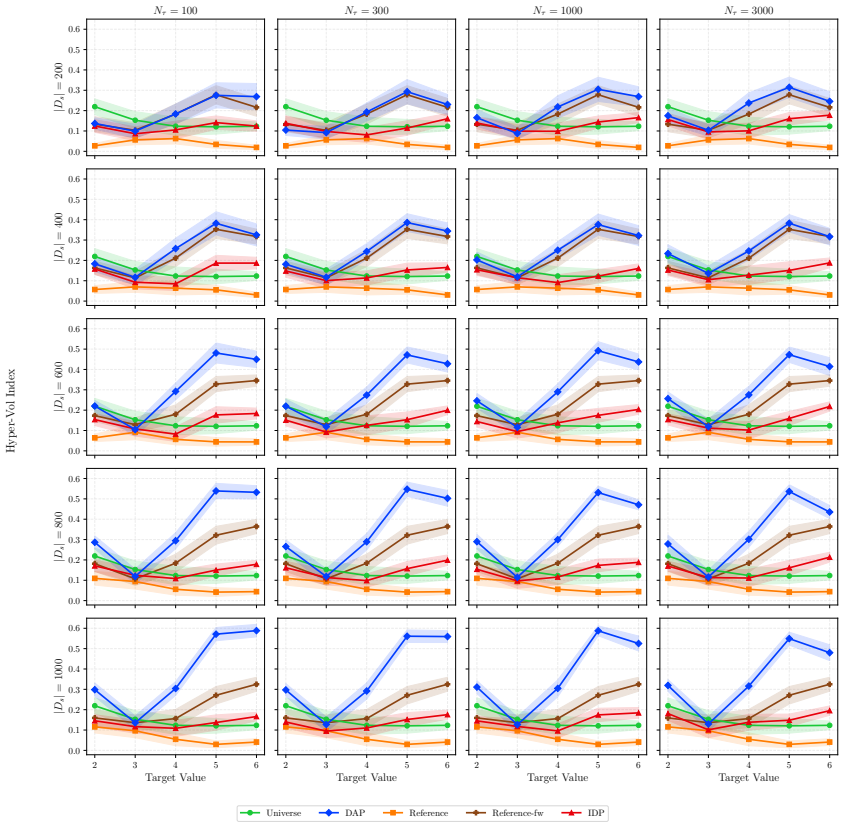


Figure 14: HVI as a function of the target lipophilicity for the QMugs benchmark under discrete flow guidance. Each column corresponds to a different number of augmented trajectories and each row to a different reference dataset size.

an anchor:

$$\begin{aligned}
\|\Phi\theta^*_{\text{anc}} - f^*\|_Q &\le \|\Phi\theta^*_{\text{anc}} - \Pi f^*\|_Q + \|\Pi f^* - f^*\|_Q && (29)\\
&= \|\Pi\mathcal{U}\,\Phi\theta^*_{\text{anc}} - \Pi f^*\|_Q + \|\Pi f^* - f^*\|_Q && (30)\\
&\le \|\mathcal{U}\,\Phi\theta^*_{\text{anc}} - f^*\|_Q + \|\Pi f^* - f^*\|_Q && (31)\\
&= \|\mathcal{U}\,\Phi\theta^*_{\text{anc}} - \mathcal{U}\,f^*\|_Q + \|\Pi f^* - f^*\|_Q && (32)\\
&\le \sigma_{\mathbf{P}^|}\,\|\Phi\theta^*_{\text{anc}} - f^*\|_Q + \|\Pi f^* - f^*\|_Q. && (33)
\end{aligned}$$

Each step: equation 29 is the triangle inequality; equation 30 substitutes the fixed-point identity from Stage 1; equation 31 uses that Π is an L^2_Q orthogonal projection and hence non-expansive ($\|\Pi v\|_Q \le \|v\|_Q$); equation 32 uses $f^* = \mathcal{U}f^*$; and equation 33 applies the contraction property of $\mathcal{U}$ in $\|\cdot\|_Q$, which we verify below.

Verification of the contraction. Let $\Delta := H_1 - H_2$. From Eq. equation 17, $\mathcal{U}H_1 - \mathcal{U}H_2 = 0$ on $\text{supp}(P_{\text{bnd}})$ and equals $\mathbf{P}^|\Delta$ on $\text{supp}(P_{\text{intr}})$. Applying Jensen to the conditional expectation and then the domination from Eq. equation 19,

$$\begin{aligned}
\|\mathcal{U}H_1 - \mathcal{U}H_2\|^2_Q = w_{\text{intr}}\,\|\mathbf{P}^|(H_1 - H_2)\|^2_{D_P} &\le w_{\text{intr}}\,\langle (H_1 - H_2)^2,\, P_{\text{intr}}\,\mathbf{P}^|\rangle\\
&\le \sigma^2_{\mathbf{P}^|}\,\langle (H_1 - H_2)^2,\, Q\rangle = \sigma^2_{\mathbf{P}^|}\,\|H_1 - H_2\|^2_Q, \qquad (34)
\end{aligned}$$

Using Q on the RHS is what makes this tight: the boundary mass that $\mathbf{P}^|$ leaks onto $\text{supp}(P_{\text{bnd}})$ is absorbed by D_{bnd} rather than dropped.

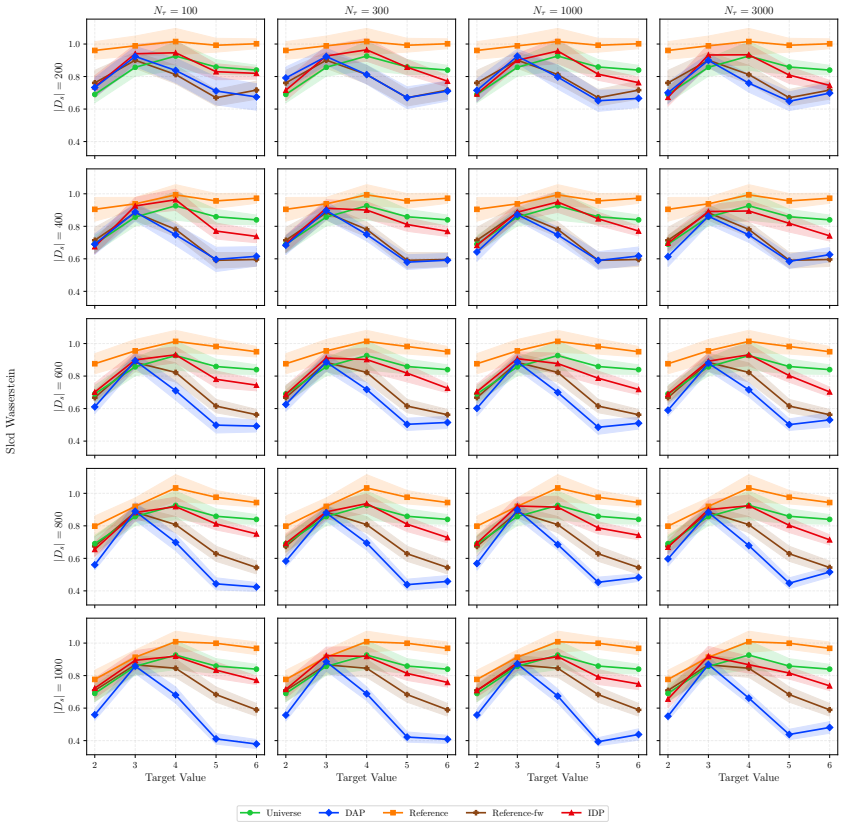


Figure 15: SWD ratio as a function of the target lipophilicity for the QMugs benchmark under discrete flow guidance. Each column corresponds to a different number of augmented trajectories and each row to a different reference dataset size.

Rearranging Eq. equation 33 under (A3) gives

$$\|\Phi\theta^*_{\text{anc}} - f^*\|_Q \;\leq\; \frac{1}{1-\sigma_{\mathbf{P}^|}}\,\|\Pi f^* - f^*\|_Q, \tag{35}$$

which, squared on both sides, yields Eq. equation 22. The realizability corollary is immediate: if $f^* \in \text{span}(\Phi)$, then $\Pi f^* = f^*$ and both sides of the bound vanish. □

A.2.3 Diffusion-Specific Reading of the Assumptions

These interpretations are worth spelling out in the DAP setting.

On the role of $\sigma_{\mathbf{P}^|} < 1$. $\sigma_{\mathbf{P}^|}$ plays the role of the sub-unit discounting factor of approximate dynamic programming Tsitsiklis & Van Roy (1997); unlike the RL setting, where a discount $\gamma < 1$ is imposed by the problem, here $\sigma_{\mathbf{P}^|}$ is a data-dependent quantity set by how well the training measure Q covers the one-step push-forward $P_{\text{intr}}\mathbf{P}^|$ (Eq. equation 19). Two observations are worth making explicit. First, the two supervision sources play asymmetric roles: P_{bnd} adds mass to the denominator Q wherever the push-forward places density on $\text{supp}(P_{\text{bnd}})$, strictly lowering the ratio there, while on purely intermediate entries the ratio reduces to an adjacent-time marginal ratio of the reverse chain, which is typically close to one for a well-trained prior. Hence the boundary term not only anchors the fixed point (see the next paragraph) but also tightens the contraction. Second, $\sigma_{\mathbf{P}^|} < 1$ is therefore not automatic: it is a preconditioning requirement on the pairing of D_s with the reverse chain induced by g_u, and is thus problem specific. Under realisability (Corollary 1) the bound collapses regardless of the value of $\sigma_{\mathbf{P}^|}$, so the contraction condition affects only the tightness of the approximation bound in the misspecified regime, not the existence of the fixed point.

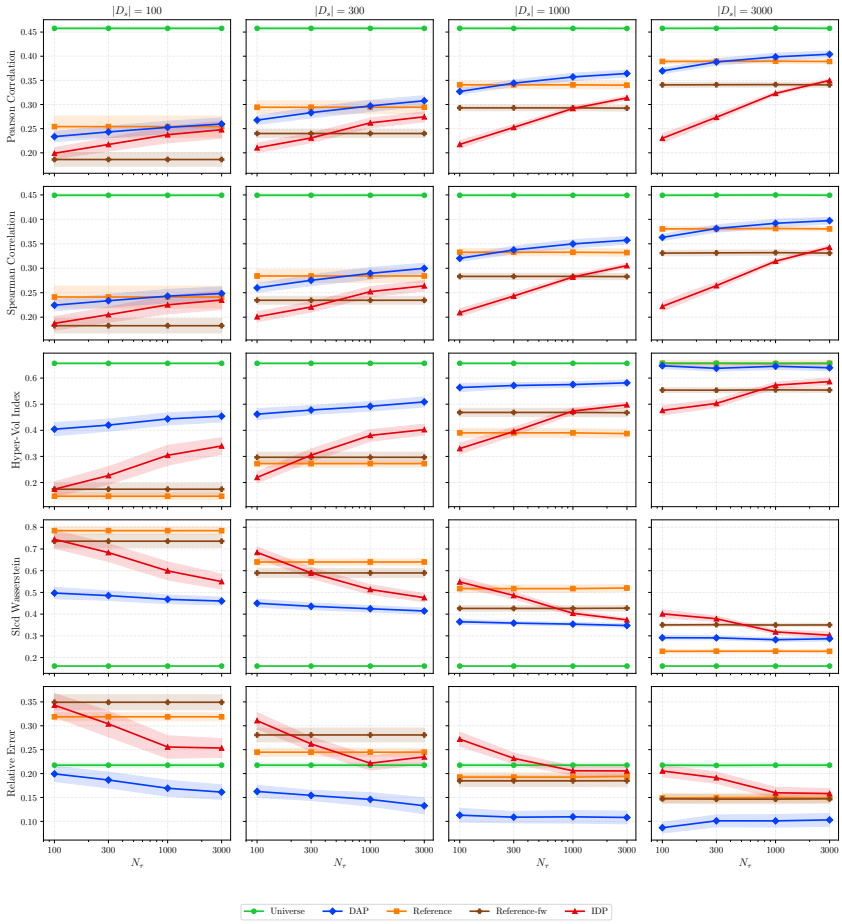


Figure 16: Prediction and steering quality as a function of the number of augmented trajectories for the QMugs benchmark under predictor guidance in the latent space of molecules, targeting a lipophilicity of five. Prediction quality is measured by Spearman and Pearson correlation coefficients averaged over the time interval $[0\text{–}100]\%$. Each column corresponds to a different reference dataset size.

Boundary loss as the anchor for the fixed point. Without the boundary term ($w_{\text{bnd}} = 0$ in Eq. equation 2), the fixed-point system Eq. equation 20 reduces to the homogeneous equation $\Phi\theta = \mathbf{P}^{\downarrow}\,\Phi\theta$, which admits every constant function $f \equiv c$ as a solution: the inter-step constraint is scale- and shift-invariant along a trajectory. Any such constant is a fixed point of $\mathcal{U}$, and the anchored-target iterates would converge to whichever one the initialization happens to sit closest to in $\|\cdot\|_Q$. The boundary loss breaks this invariance by supplying a nonzero $\mathbf{b} = \Phi^\top D_{\text{bnd}} Y_{\text{bnd}}$: it pins the solution at $t = 0$ to the labelled values, and the inter-step constraint propagates these pinned values backward in time along the reverse chain. In the bound Eq. equation 22, this role is visible in the mixed norm $\|\cdot\|_Q$: the projection error is controlled jointly on the labelled boundary and on the intermediate trajectory, with relative weight w_{intr}. This is the structural mechanism by which a small labelled set is amplified by the reverse dynamics of a rich prior to calibrate the oracle everywhere along the trajectory.

A.3 Anchored vs Non-Anchored Targeting

The single-sample estimator of Eq. equation 5, optimized without gradient detachment, is a consistent estimator not of $\mathcal{L}_{\text{intr}}$ but of $\mathcal{L}_{\text{intr}} + \mathbb{E}_{\tau,t}[\mathrm{Var}[f_\theta(x_{t-1}) \mid x_t]]$ (Eq. equation 6). The extra term is a θ-dependent penalty on the conditional variance of f_θ along the reverse chain, and it enters the gradient

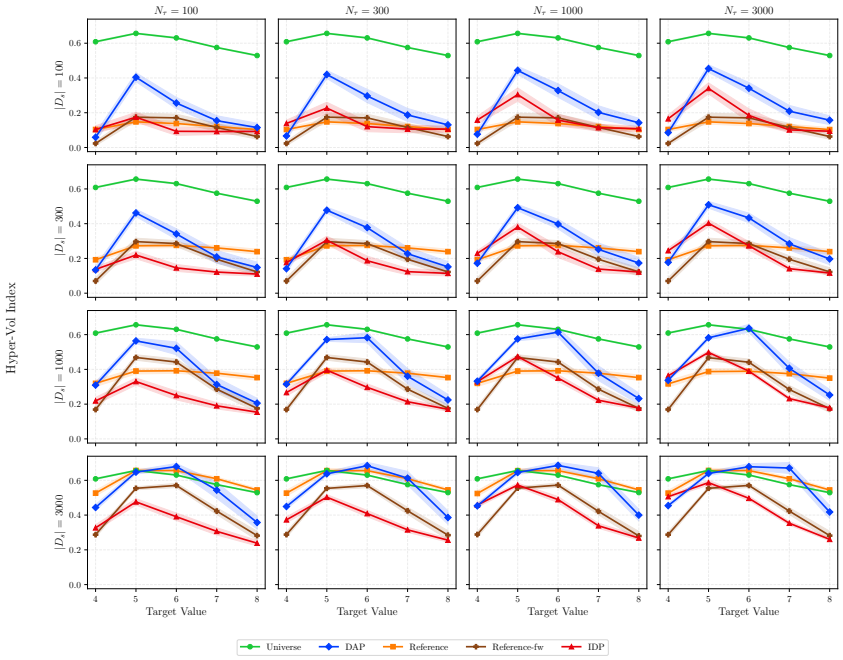


Figure 17: HVI as a function of the target lipophilicity for the QMugs benchmark under predictor guidance in the latent space of molecules. Each column corresponds to a different number of augmented trajectories and each row to a different reference dataset size.

at every step. Its minimizers are *flat* predictors in a precise sense: functions whose output varies little between consecutive reverse-diffusion states, with the constant function as the limiting case that zeroes the penalty outright and trivially smooths the reverse dynamics away. The non-anchored variant therefore converges to a stationary point of this regularized surrogate, not of the inter-step constraint; by contrast, the anchored target removes the penalty exactly (Proposition 1) and recovers the true solution of the constraint under realizability (Corollary 1).

A.4 The Non-Linear Case and the Role of the EMA Window

Theorem 1 is stated for the linear class of Eq. equation 13, whereas the oracles used in our experiments are MLPs and CNNs. The EMA target is itself what addresses this gap. In the large-window limit $\alpha \to 1$ the target moves so slowly that it is effectively frozen, and with a frozen target the anchored intermediate loss of Eq. equation 9 reduces to ordinary supervised regression onto fixed labels, which converges for any differentiable function class. This is the regime that assumption (A5) formalizes: the two-timescale stochastic approximation framework of Borkar (1997) applies to non-linear as well as linear parameterizations, and it is precisely why a slow-moving target is expected to stabilize non-linear training.

The EMA window $1/(1-\alpha)$ therefore trades stability against optimization speed. Enlarging it buys stability, since the target is closer to frozen and the effective problem closer to plain regression, but it also requires more iterations to converge, since the target must still catch up with θ. The ablation in Appendix A.12 shows this trade-off is comfortably practical: prediction and guidance quality remain stable across window sizes from 25 to 100 (Figures 20 and 21) without an impractical training budget, and degradation appears only at larger windows. If instability is a concern in a new setting, enlarging the window is thus the natural remedy.

A.5 Alternative Unbiased Loss Estimators

An alternative approach to removing the bias in Eq. equation 6 is double sampling. Drawing two independent samples $\tilde{x}_{t-1}^{(1)}, \tilde{x}_{t-1}^{(2)} \sim p(x_{t-1}|x_t^\tau)$ and pairing them in the residual defines the double-

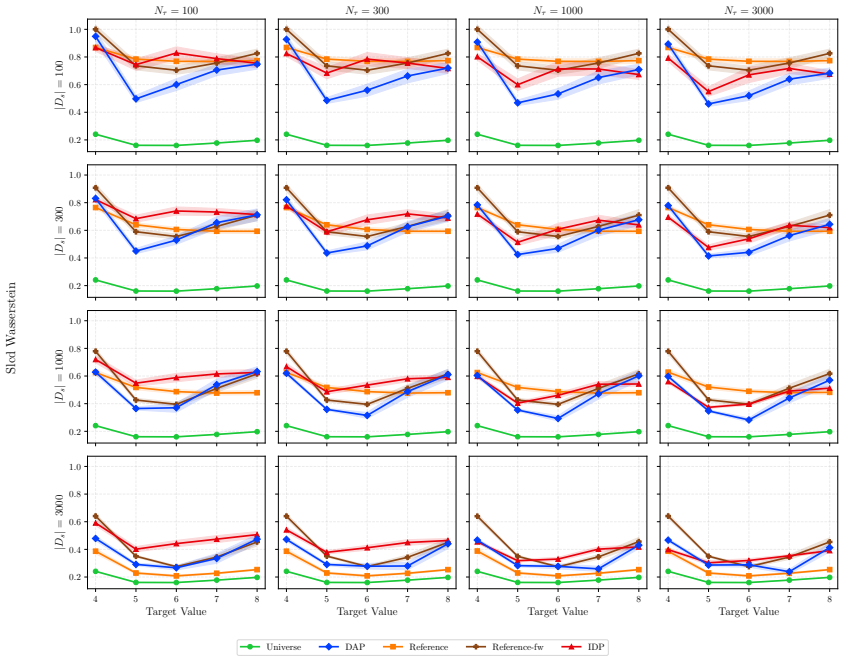


Figure 18: SWD ratio as a function of the target lipophilicity for the QMugs benchmark under predictor guidance in the latent space of molecules. Each column corresponds to a different number of augmented trajectories and each row to a different reference dataset size.

sampling estimator

$$\hat{\mathcal{L}}_{\text{intr}}^{\text{dbl}} = \mathbb{E}_{\tau,t}\Big[\big(f_\theta(x_t^\tau) - f_\theta(\tilde{x}_{t-1}^{(1)})\big)\big(f_\theta(x_t^\tau) - f_\theta(\tilde{x}_{t-1}^{(2)})\big)\Big]. \tag{36}$$

Independence across the two samples removes the variance cross-term, contrasting cleanly with the biased single-sample estimator:

$$\mathbb{E}\big[\hat{\mathcal{L}}_{\text{intr}}^{\text{dbl}}\big] = \mathcal{L}_{\text{intr}}. \tag{37}$$

This unbiasedness comes at a price. Beyond the doubled posterior-sampling cost at each step, double sampling delivers full-gradient optimization of $\mathcal{L}_{\text{intr}}$, and full-gradient methods of this form have been repeatedly observed to underperform their semi-gradient counterparts in related settings Saleh & Jiang (2019); Fujimoto et al. (2022); Yin et al. (2022); Chen et al. (2021), including in deterministic settings where the double-sampling trick collapses to a single sample and the argument is not one of statistical efficiency. Intuitively, partially detaching the gradient forfeits some control over the descent direction and, in exchange, appears to avoid failure modes that afflict full-gradient residual minimization Baird (1995).

A.6 Cross-Entropy Lower Bound at Intermediate Time Steps

CE lower bound. The CE is averaged over all trajectories τ at time t, which by the law of total expectation decomposes as $\mathbb{E}_\tau[\text{CE}(\cdot)] = \mathbb{E}_{x_t}[\mathbb{E}_{\tau|x_t}[\text{CE}(\cdot)]]$. For a trajectory τ at x_t, the per-trajectory CE is $\text{CE}(\mathbf{1}_{y=y_0^\tau}, p_\theta(y|x_t))$. Averaging over trajectories for a fixed x_t, and using the linearity of CE in its first argument together with the fact that $p_\theta(y|x_t)$ does not depend on τ given x_t,

$$\mathbb{E}_{\tau|x_t}\big[\text{CE}(\mathbf{1}_{y=y_0^\tau}, p_\theta(y|x_t))\big] = \text{CE}\big(\mathbb{E}_{\tau|x_t}[\mathbf{1}_{y=y_0^\tau}],\, p_\theta(y|x_t)\big) = \text{CE}(p(y|x_t),\, p_\theta(y|x_t))\,, \tag{38}$$

where the last equality uses $\mathbb{E}_{\tau|x_t}[\mathbf{1}_{y_k=y_0^\tau}] = p(y_k|x_t)$. Averaging over x_t, the objective becomes $\min_\theta \mathbb{E}_{x_t}[\text{CE}(p(y|x_t), p_\theta(y|x_t))]$, which is minimised if and only if $p_\theta(y|x_t) = p(y|x_t)$. □

CE at $t = 0$. At $t = T$, x_T is pure noise and carries no information about y, so $p(y|x_0) = \text{Uniform}(1/K)$ over K classes, and the CE equals $\log K$, the entropy of the uniform distribution.

A.7 Oracle Summary and Experimental Grid

Experimental variables. We vary the size of the reference dataset $|D_s|$ and the number of augmented trajectories N_τ to characterise oracle performance across a wide range of data regimes. Since IDP relies on pseudo-labels produced by f_s, its performance is sensitive to the generalisation quality of the reference oracle. To study this sensitivity in isolation, we additionally evaluate oracles under three label perturbation schemes (uniform, shuffle, and adversary) applied to f_s at varying perturbation extents; full descriptions are provided in Appendix A.9.

Table 2 provides an overview of the four benchmarks. Table 3 summarises the four oracle types, their training data, and the space in which they operate. Tables 4 and 5 summarise the arrays of experimental variables explored for each benchmark.

A.8 Evaluation Details

Prediction quality evaluation. All prediction metrics are computed on held-out reverse-diffusion trajectories (500 for halfmoon and MNIST; 1000 for molecular benchmarks): accuracy and cross-entropy for classification tasks, Spearman and Pearson correlation for regression tasks. In plots showing a metric as a function of time, each value at percentile t represents the metric averaged over the interval $[t\text{–}100]\%$ across all held-out trajectories. When reported as a scalar (e.g., as a function of $|D_s|$, N_τ, or perturbation extent), the metric is averaged over the full interval $[0\text{–}100]\%$.

Steering quality evaluation. For each configuration and random seed, we generate guided and unguided trajectories (500 each for halfmoon and MNIST; 1000 each for molecular benchmarks). In classification tasks, steering quality is assessed by comparing the ability of guided and unguided generation to produce target-class samples. In regression tasks, steering is directed towards a continuous target value y_{trg} of the desired molecular property.

Reward. *Classification.* The universe oracle f_u, trained on the full universe dataset, serves as the ground truth evaluator: the reward for each generated sample is f_u's predicted probability of the target class. The mean reward $\bar{R}$ is averaged over all trajectory endpoints, and $\Delta\bar{R} = \bar{R}_{\text{guided}} - \bar{R}_{\text{prior}} \in [0, 1]$ captures the improvement attributable to guidance. This ground truth reward is used only for evaluation, not for steering; the steering reward is the target-class probability predicted by the oracle used for guidance. $\Delta\bar{R}$ is the metric shown in all steering plots associated with classification (halfmoon and MNIST benchmarks).

Regression. For evaluation, RDKit computes the molecular property (lipophilicity or number of rings) of each generated sample. We average the RDKit-derived values over guided samples to obtain $\bar{y}_{\text{gd}}$ and report the relative error $E = |\bar{y}_{\text{gd}} - y_{\text{trg}}|/|y_{\text{trg}}|$. Relative error is the metric shown in all steering plots associated with regression.

For steering, all molecular oracles follow the design of Nisonoff et al. (2024): each oracle models the mean and standard deviation of the property, and the steering reward is the likelihood of y_{trg} under the oracle's predicted normal distribution.

SWD computation. We compute the Sliced Wasserstein Distance (SWD) between the generated samples and a universe target set to measure distributional similarity. SWD averages one-dimensional Wasserstein distances over random projections, providing a tractable approximation to the full Wasserstein distance.

For classification, the universe target set is all target-class samples from D_u. For regression, it comprises all universe samples whose property value lies within $y_{\text{trg}} \pm \delta$, where $\delta = 0.1$ for discrete flow guidance and $\delta = 0.5$ for latent-space guidance molecular benchmarks. Both the generated and universe target sets are embedded using the feature backbone of f_u.

Prior to computing SWD, both embedded sets are normalised by a scale factor derived from the universe target embeddings X_{tgt}: scale $= \sqrt{\sum_d \text{Var}_d[X_{\text{tgt}}]}$, the square root of the total variance summed over all embedding dimensions. This normalisation ensures that a point mass at the universe target mean yields an SWD of approximately one, providing an interpretable reference. Since the generated set is smaller than the universe target set, the universe target set is subsampled to the generated set size via percentile interpolation before computing SWD.

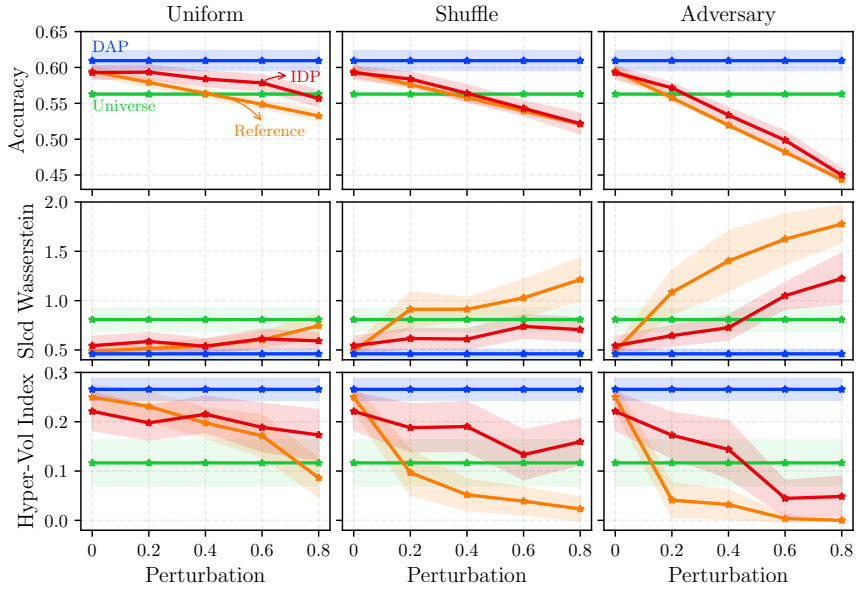


Figure 19: **DAP is invariant to pseudo-label perturbation while IDP degrades monotonically, demonstrating robustness to an imperfect reference oracle.** Perturbation simulates a reference oracle that does not generalise beyond its training set. Accuracy, SWD ratio, and HVI as a function of perturbation extent for the uniform, shuffle, and adversary schemes ($|D_s| = 100$, $N_\tau = 30$).

The SWD ratio is defined as $\mathrm{SWD}(\text{guided}, \text{target})/\mathrm{SWD}(\text{prior}, \text{target})$; a value of zero indicates perfect match to the target distribution, and values above one indicate degradation relative to the prior. In classification, the prior samples uniformly across classes, so guided and prior SWDs should differ under effective guidance. In regression, however, when y_{trg} lies near the median of the generative model's training distribution, the prior already produces samples close to the target set, yielding an SWD ratio near one even under effective guidance. Our benchmark evaluates steering quality across a range of target property values, including those near the training distribution median.

HVI. For classification, $\text{HVI} = \Delta\bar{R} \times (1 - \text{SWD ratio})$, with contributions zeroed out when $\Delta\bar{R} < 0$ or SWD ratio > 1. For regression, $\text{HVI} = (1 - E) \times (1 - \text{SWD ratio})$, with contributions zeroed out when $E > 1$ or SWD ratio > 1. In both cases, $\text{HVI} \in [0, 1]$.

Shaded bands in all line plots show 95% t-distribution confidence intervals across seeds. In all figures, the y-axis label *Slcd Wasserstein* refers to the SWD ratio as defined above.

Examples of samples generated by steering with different oracles are shown in Figures 9 (MNIST), 30 and 32 (discrete flow guidance), and 34 (latent-space guidance).

A.9 DAP is Robust to Pseudo-Label Quality

In challenging real-world applications, the reference dataset is a limited representation of the universe and the reference oracle may not generalise well outside its training set; label perturbation is used here to simulate this failure mode in a controlled manner. Three perturbation schemes are applied to the output of the reference oracle f_s at varying perturbation probabilities $p \in [0, 1]$. With probability $1 - p$ the oracle output is unchanged; with probability p the output is modified as follows. **Uniform**: the oracle outputs a uniform distribution over all classes. **Shuffle**: the oracle outputs its predicted class probabilities in a randomly shuffled order. **Adversary**: the oracle outputs a random class other than the one with the highest predicted probability. The three schemes are listed in order of increasing aggressiveness.

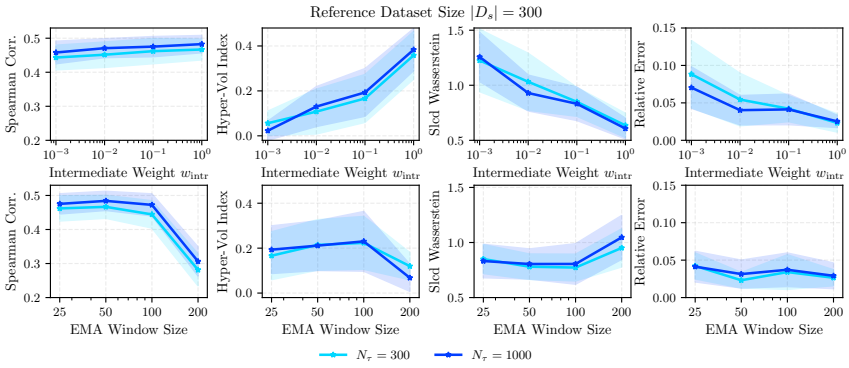


Figure 20: Ablation of DAP hyperparameters on the QMugs molecule benchmark under discrete flow guidance, at reference dataset size $|D_s| = 300$. *Top*: prediction and steering quality as w_{intr} varies, with the EMA window size fixed at 25. *Bottom*: the same metrics as the EMA window size varies, with w_{intr} fixed at 0.1.

Figure 19 studies oracle robustness when the reference oracle (f_s) is degraded via these three schemes at varying extents, for reference dataset size $|D_s| = 100$ and number of augmented trajectories $N_\tau = 30$. As perturbation extent increases, IDP's prediction accuracy and steering quality degrade monotonically, since its intermediate supervision relies directly on pseudo-labels produced by f_s. In contrast, DAP's performance is invariant to perturbation extent across all three schemes, as its training does not involve labelling the augmented trajectories. The degradation is most pronounced under the adversary scheme and least under uniform perturbation, consistent with the increasing aggressiveness of the perturbation schemes.

The same observations hold across all $|D_s|$, N_τ, and perturbation schemes, where DAP's prediction and steering quality remain unchanged, while IDP's deteriorate with perturbation extent. Furthermore, IDP's accuracy and cross-entropy as a function of reverse time step degrade as perturbation extent increases, while DAP's time-resolved prediction quality is unaffected.

A.10 Implementation Details

Universe datasets. The half-moon universe dataset consists of 50,000 samples across 2 classes. The MNIST universe dataset uses the standard 70,000-sample split (60,000 train / 10,000 test) across 10 classes. In both cases the reference dataset D_s is sampled from the training split with equal samples per class (stratified).

The molecular benchmarks use the processed QMugs dataset Nisonoff et al. (2024), consisting of drug-like molecules derived from ChEMBL, following the same setup as the DFM guidance work that introduced this benchmark. The DFM guidance benchmark uses the full dataset; the latent-space guidance benchmark uses a 50K subset.

Oracle architectures. For half-moon, the time-conditioned oracle (DAP, IDP) is an MLP with hidden layers $[64, 32]$ (ReLU, dropout 0.1), taking the 2-D state x_t as input, with 2-class output and time conditioning. For MNIST, it is a CNN with two convolutional layers ($1 \to 32 \to 64$ channels, 3×3 kernels, each followed by ReLU and $2\times$ max-pooling) and a linear output head (10 classes), with time conditioning. The reference and universe oracles (f_s, f_u) use the same backbone without time conditioning.

For the DFM molecular benchmark, all oracles are 2-layer MLPs over one-hot flattened SMILES tokens. Time-conditioned oracles (f_{DAP}, f_{IDP}, f_{fw}) concatenate the scalar t to the input, following Nisonoff et al. (2024).

For the latent-space molecular benchmark, f_u and f_s are 2-layer MLPs over soft-SELFIES representations without time conditioning. Time-conditioned oracles (f_{DAP}, f_{IDP}, f_{fw}) augment the same MLP with adaLN at each hidden layer, conditioned on $\log(\sigma_t)$ via sinusoidal embedding.

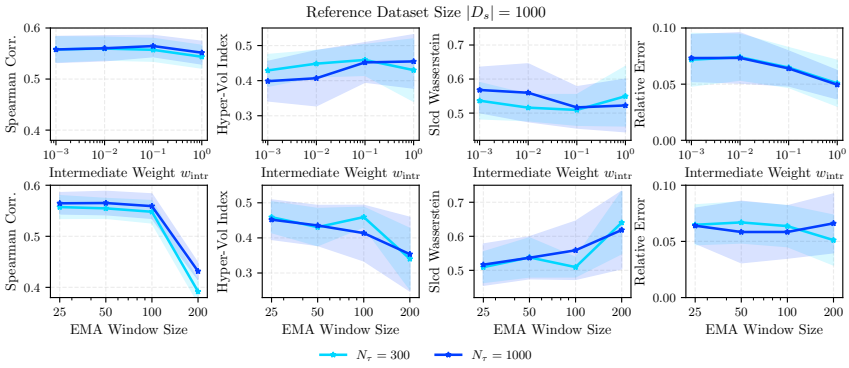


Figure 21: Ablation of DAP hyperparameters on the QMugs molecule benchmark under discrete flow guidance, at reference dataset size $|D_s| = 1000$. *Top*: prediction and steering quality as w_{intr} varies, with the EMA window size fixed at 25. *Bottom*: the same metrics as the EMA window size varies, with w_{intr} fixed at 0.1.

Discrete Flow Model (DFM) and the Variational Auto Encoder (VAE) We trained a discrete diffusion transformer (DDiT) via masked discrete flow matching on the SMILES representation of QMugs molecules (8 blocks, 320 hidden dim, 10 attention heads, AdaLN timestep conditioning, RoPE) Lipman et al. (2024). Sampling uses 200 denoising steps without stochasticity (i.e., without the ability to mask a previously unmasked token), yielding $\sim$50% valid molecules. By comparison, the MLP-based discrete flow model of Nisonoff et al. (2024) produces only $\sim$13% valid molecules at 1000 steps with 30% stochasticity; at matching steps and stochasticity, our model reaches $\sim$80% validity under the same criteria. We trained a stronger generative model to enable a fair comparison across all oracles.

For the latent-space benchmark, we use an MLP-based VAE operating on SELFIES representations (latent dim 1024), with the architecture of Eckmann et al. (2022) and a pretrained checkpoint from Oestreich et al. (2025). A VP-SDE score network is trained on the VAE latent space using the universe dataset.

DAP hyperparameters. A single fixed set of hyperparameters was used for DAP across all configurations on each dataset, with no per-configuration tuning. For half-moon, $w_{\text{intr}} = 0.1$ and the EMA target network window is 100. For MNIST, $w_{\text{intr}} = 0.001$ and the EMA window is 100. For the discrete flow molecular benchmark, $w_{\text{intr}} = 0.1$ and the EMA window is 25. For the latent-space molecular benchmark, $w_{\text{intr}} = 0.01$ and the EMA window is 25.

Reverse sampling uses 100 steps for half-moon, 200 steps for MNIST, 200 steps for the discrete flow molecular benchmark, and 100 steps for the latent-space molecular benchmark. FK steering was

Table 6: Ablation over the intermediate loss weight w_{intr} on Half-Moon ($|D_s|$=100, N_τ=10). Upon tuning w_{intr}, DAP performance surpasses all oracles. The reference oracle is shown as it had the best performance among all other oracles in this setting. Accuracy is averaged over time [0–100]%. All values are mean $\pm$ std over 20 seeds.

Oracle	w_{intr}	Accuracy	$\Delta\bar{R}$	Slcd Wasserstein	Hyper-Vol Index
DAP	0.001	0.58 ± 0.03	0.44 ± 0.06	0.48 ± 0.13	0.23 ± 0.06
DAP	0.003	0.58 ± 0.03	0.46 ± 0.06	0.46 ± 0.11	0.24 ± 0.06
DAP	**0.01**	0.59 ± 0.03	0.48 ± 0.04	0.48 ± 0.22	$\mathbf{0.26 \pm 0.08}$
DAP	0.03	0.59 ± 0.03	0.45 ± 0.05	0.47 ± 0.16	0.24 ± 0.08
DAP	0.1	0.59 ± 0.02	0.46 ± 0.06	0.60 ± 0.21	0.19 ± 0.09
DAP	0.3	0.59 ± 0.02	0.43 ± 0.09	0.51 ± 0.14	0.21 ± 0.07
Reference	—	0.59 ± 0.02	0.49 ± 0.04	0.49 ± 0.11	0.25 ± 0.05

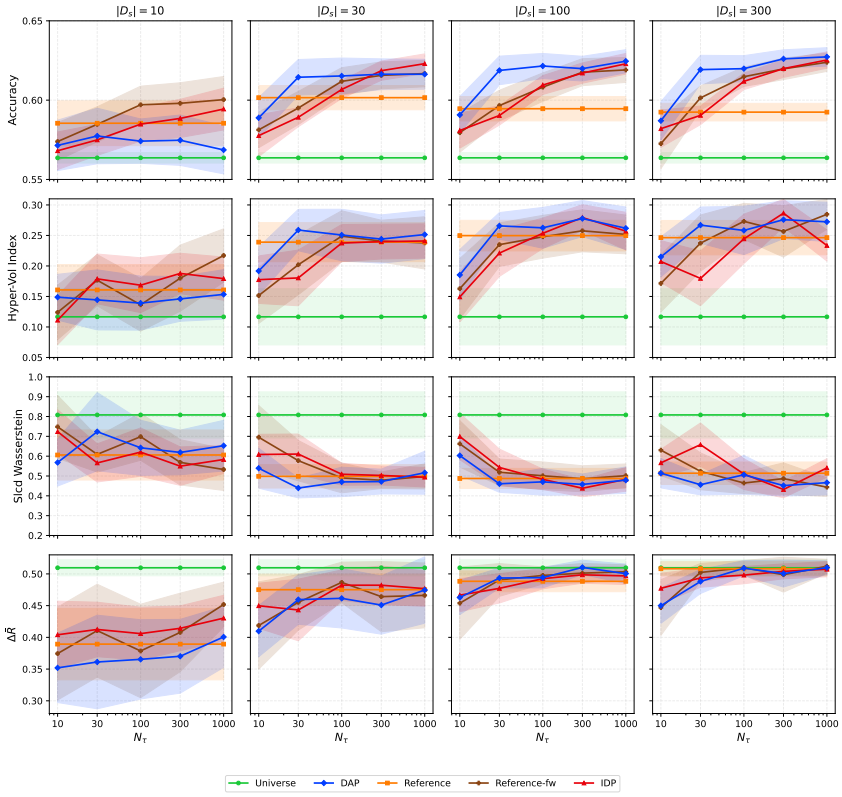


Figure 22: Accuracy and steering quality metrics as a function of the number of augmented trajectories for the Half-Moon dataset. Accuracy is averaged over the entire time interval $[0\text{–}100]\%$. Each column corresponds to a different reference dataset size.

applied under fixed parameters across all half-moon and MNIST settings; all experiments use the difference potential with $\lambda = 5.0$ and resampling at every step.

For the discrete flow molecular benchmark, we use the gradient-based predictor guidance for discrete flow models developed by Nisonoff et al. (2024). For the latent-space molecular benchmark, we use classifier guidance Song et al. (2020); Dhariwal & Nichol (2021) on the VAE latent space. In both molecular benchmarks, we select the best guidance hyperparameter per oracle: the guidance temperature for discrete flow guidance, and the gradient weight for classifier guidance.

A.11 Loss Balance at Data Extremes

Since a single fixed set of hyperparameters was used across all configurations (Appendix A.10), performance degradation at the extremes of very small $|D_s|$ or N_τ is expected. DAP's objective balances a supervised boundary loss on D_s with an intermediate consistency loss over augmented trajectories (weighted by w_{intr}); stable performance requires these two terms to remain commensurate. With very small $|D_s|$, a large w_{intr} overpowers the limited boundary supervision, causing the oracle to lose its grounding in the original label space. With very small N_τ, the same weight amplifies noise from the sparse trajectory sample at the expense of the supervised signal. Per-configuration tuning of w_{intr} would be expected to recover performance. As shown in Table 6, the guidance performance of DAP upper-bounds that of other oracles upon tuning of w_{intr}, as studied on Half-Moon. Note that the main benchmark of the paper was carried out with a higher value of w_{intr} than the best-performing setting.

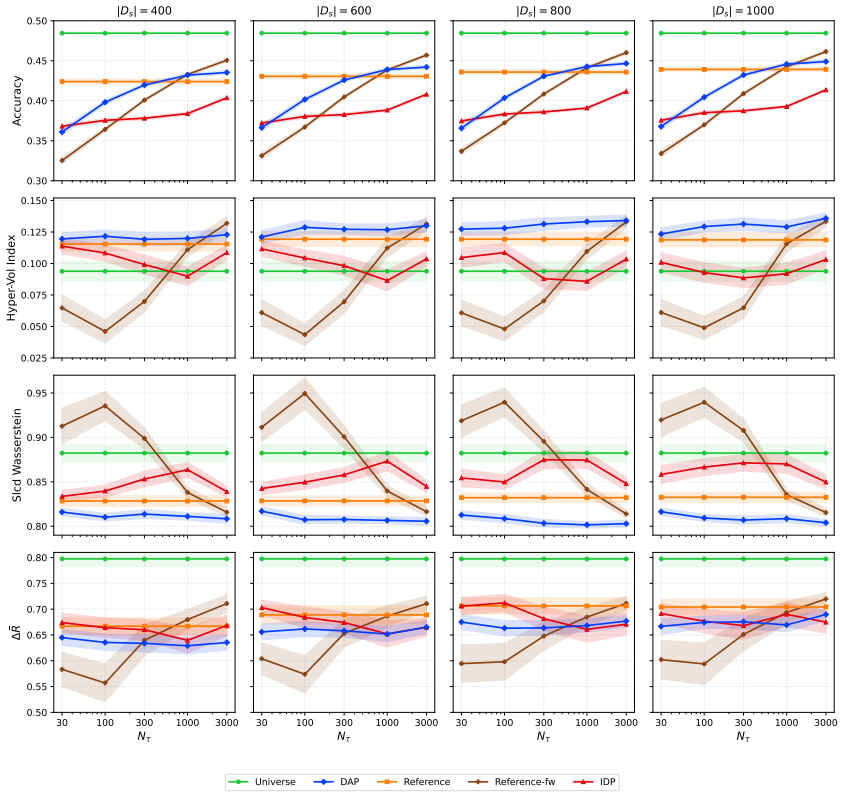


Figure 23: Accuracy and steering quality metrics as a function of the number of augmented trajectories for the MNIST dataset. Accuracy is averaged over the entire time interval $[0\text{–}100]\%$. Each column corresponds to a different reference dataset size.

A.12 Hyperparameter Study: EMA Window Size and Intermediate Weight

We additionally ablate the impact of two hyperparameters, w_{intr} and the EMA window size $1/(1-\alpha)$, on DAP's prediction and guidance performance. The ablation was conducted on discrete flow guidance for molecules steering towards lipophilicity, one of our more challenging molecular benchmarks, for generality. We sweep over the joint grid of $w_{\text{intr}} \in \{0.001, 0.01, 0.1, 1\}$ and EMA window sizes $\in \{25, 50, 100, 200\}$, across all configurations of reference dataset sizes $|D_s| \in \{300, 1000\}$ and number of trajectories $N_\tau \in \{300, 1000\}$.

For a fixed intermediate loss weight w_{intr}, prediction and guidance quality remain stable across a wide range of EMA window sizes (25 to 100). Larger window sizes deteriorate performance, since the target moves too slowly to converge within the same fixed training budget.

At any fixed EMA window size, changing w_{intr} has a more pronounced impact on guidance performance (HVI) at a smaller reference dataset size (Figure 20) than at a larger one (Figure 21). Prediction performance, averaged over time steps and evaluation trajectories, stays relatively robust to $10^3\times$ changes in w_{intr}.

Our molecule benchmark, conducted on a discrete flow model, used $w_{\text{intr}} = 0.1$ throughout, while this ablation shows that $w_{\text{intr}} = 1$ performs best. Since running the full benchmark grid at the ablation-optimal w_{intr} was computationally prohibitive, we used a single fixed set of hyperparameters across all configurations; the reported results therefore represent a conservative estimate of DAP's performance.

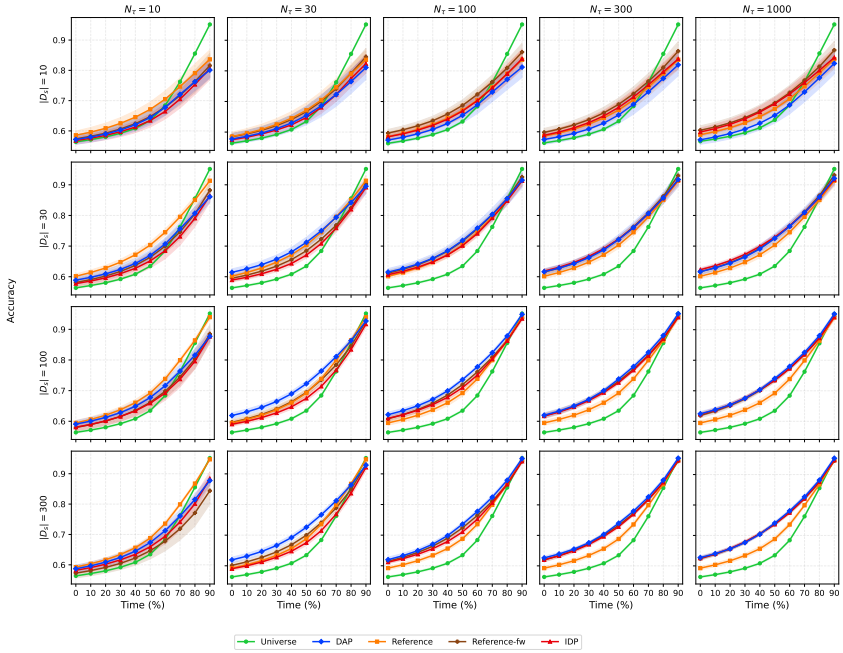


Figure 24: For all reference dataset sizes (rows) and augmented trajectory counts (columns), accuracy is plotted as a function of time on Half-Moon. Accuracy is averaged over the interval $[t\text{–}100]\%$ across evaluation trajectories.

A.13 Training Budget of the Forward-Noised Oracle

In all molecular benchmarks, the forward-noised oracle was trained with an unlimited training sample budget. This is inherent to the forward-noising process, which can generate noised samples at any continuous time value. By contrast, DAP and IDP operate on a more limited budget, since reverse sampling trajectories provide states only at discretised time points. Therefore, for Half-moon and MNIST benchmarks we studied forward-noised oracle under a similar, but not strictly equal, budget to DAP and IDP.

Concretely, only a subset of reference samples equal in size to the number of reverse augmented trajectories is forward-noised, though at any continuous time t rather than only at the discretised values used for reverse sampling. Even under this constraint, the forward-noised oracle sees more unique training pairs, since each sample can be noised at arbitrary times.

All appendix plots showing prediction and steering quality of the forward-noised oracle on these datasets were produced under this setting (Figures 22–28).

The gap between the steering quality of DAP and the forward-noised oracle decreases as the number of augmented trajectories, i.e., the subset of labelled reference samples available for noising, exceeds the reference dataset size (see Figures 22 and 23). This suggests that under a limited sample budget, the reverse dynamics exploited by DAP capture information that forward noising alone does not.

A.14 Related Work

Guidance approaches for diffusion generative models fall into three categories. **(i) Conditional training**: the generative model is trained with property labels for training points, rarely feasible in scientific settings where labels are expensive to obtain, and not a versatile strategy: as the property of interest changes across applications, the model must be retrained from scratch for each new label, making this approach prohibitively expensive at scale Ho & Salimans (2022); Yuan et al. (2023). **(ii) Fine-tuning and RL-based approaches**: the generative model's parameters are updated to maximise a reward function via approaches ranging from direct fine-tuning to value-function

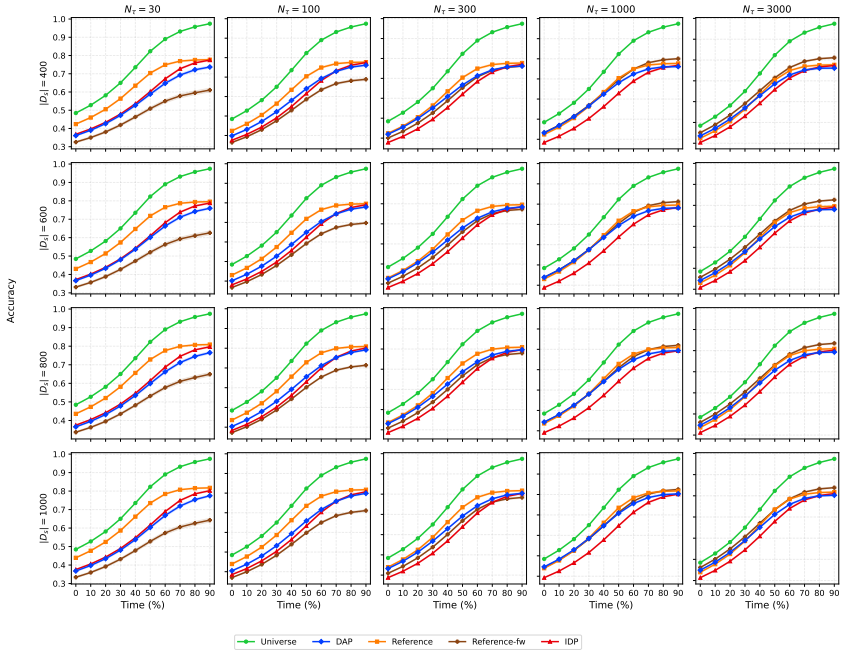


Figure 25: For all reference dataset sizes (rows) and augmented trajectory counts (columns), accuracy is plotted as a function of time on MNIST. Accuracy is averaged over the interval $[t\text{–}100]\%$ across evaluation trajectories.

optimisation Xu et al. (2023); Black et al. (2023); Domingo-Enrich et al. (2024); Wallace et al. (2024); Gao et al. (2025); in addition to requiring abundant labelled data, this couples the model to a specific task and demands dedicated training resources, rendering reuse across tasks costly. **(iii) Inference-time steering**: at generation time, a reward or oracle steers the sampling process without modifying the generative model's parameters, preserving the pretrained model and allowing it to be redirected toward any property of interest simply by swapping the oracle, the only category that is both versatile across tasks and viable when labelled data is scarce. Methods in this category split into gradient-based approaches, which augment the score function with the gradient of a reward Song et al. (2020); Dhariwal & Nichol (2021); Bansal et al. (2023); Chung et al. (2022); Janati et al. (2024); Lu et al. (2023), and particle-based approaches, which use sequential Monte Carlo or Feynman-Kac formulations to reweight particles during generation and accommodate generic black-box reward functions without requiring reward gradients Singhal et al. (2025); Trippe et al. (2022); Wu et al. (2023); Dou & Song (2024); Zhao et al. (2025); Cardoso et al. (2023); Li et al. (2024).

We target category (iii). Within this category, however, existing methods critically depend on the quality of the property oracle used during generation, yet oracle training has received comparatively little attention, especially in the low-data regime. Our contribution is not a new steering algorithm: **we propose a new way to train oracles that is specifically suited to the data-scarce setting**, exploiting reverse-trajectory augmentation and inter-step dynamics constraints derived from the generative dynamics. This is a key factor in differentiating our work from prior work such as Li et al. (2024), which relies on directly mapping intermediate states to the original space to obtain an estimate of the endpoint reward.

Because the oracle is a modular component that can be improved independently of the steering algorithm, inference-time steering methods stand to benefit from predictions better calibrated to the generative dynamics, as our experiments demonstrate. We evaluate oracle quality under FK-based steering Singhal et al. (2025), which supports arbitrary reward functions (differentiable or otherwise), and gradient-based predictor guidance Song et al. (2020); Dhariwal & Nichol (2021); both apply to discrete and continuous-state models, making them broadly applicable testbeds for assessing oracle quality.

Anchored targets under function approximation. Bertsekas & Tsitsiklis (1996) and Tsitsiklis & Van Roy (1997) analyzed approximate dynamic programming under linear function approximation; Sutton (1984) established temporal-difference learning as a central instance; Dayan (1992); Tsitsiklis & Van Roy (1996) further characterized its convergence properties. Baird (1995); Bertsekas (1995); Boyan & Moore (1995); Gordon (1995) documented divergence and instability under nonlinear approximation. Mnih et al. (2015) introduced fixed target networks for deep value learning; Fujimoto et al. (2018) replaced the fixed target with Polyak-averaged (EMA) updates. Robbins & Monro (1951); Kiefer & Wolfowitz (1952) established the stochastic-approximation machinery used in our convergence proof. Baird (1995) also proposed residual (full-gradient) minimization as a bias-free alternative; Saleh & Jiang (2019); Fujimoto et al. (2022); Yin et al. (2022); Chen et al. (2021) subsequently reported empirical and theoretical evidence that semi-gradient updates tend to outperform full-gradient residual minimization in related settings.

Additional Supplementary Figures.

Due to the extensive benchmarking conducted in this work, a large number of plots were generated covering every metric across all experimental configurations. A representative subset is included in the Appendix; the complete collection will be released in the accompanying code repository upon publication.

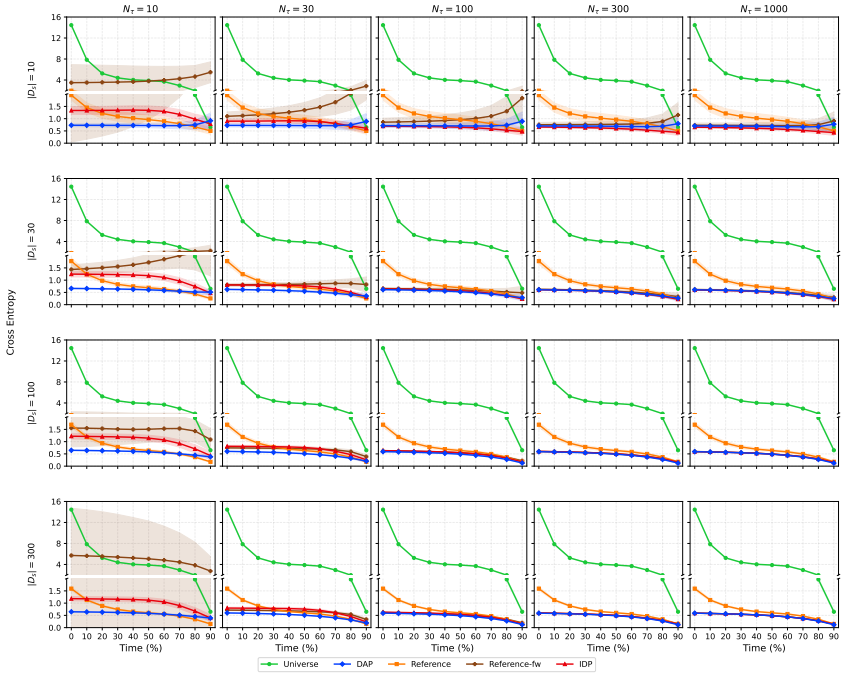


Figure 26: For all reference dataset sizes (rows) and augmented trajectory counts (columns), cross-entropy is plotted as a function of time on Half-Moon. Cross-entropy is averaged over the interval $[t\text{–}100]\%$ across evaluation trajectories.

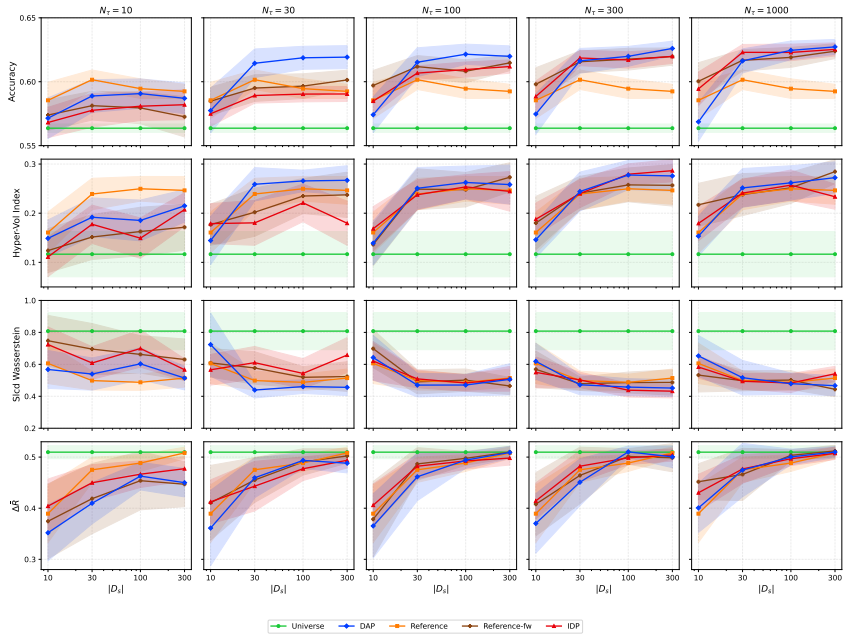


Figure 27: Accuracy and steering quality metrics as a function of the reference dataset size for the Half-Moon dataset. Accuracy is averaged over the entire time interval $[0\text{–}100]\%$. Each column corresponds to a different number of augmented trajectories.

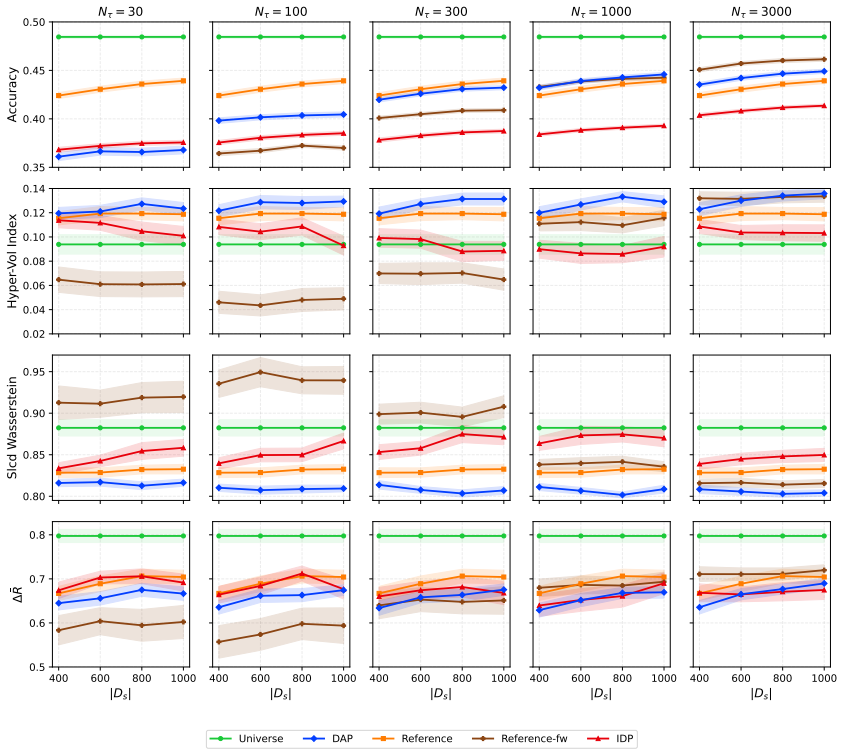


Figure 28: Accuracy and steering quality metrics as a function of the reference dataset size for the MNIST dataset. Accuracy is averaged over the entire time interval $[0\text{–}100]\%$. Each column corresponds to a different number of augmented trajectories.

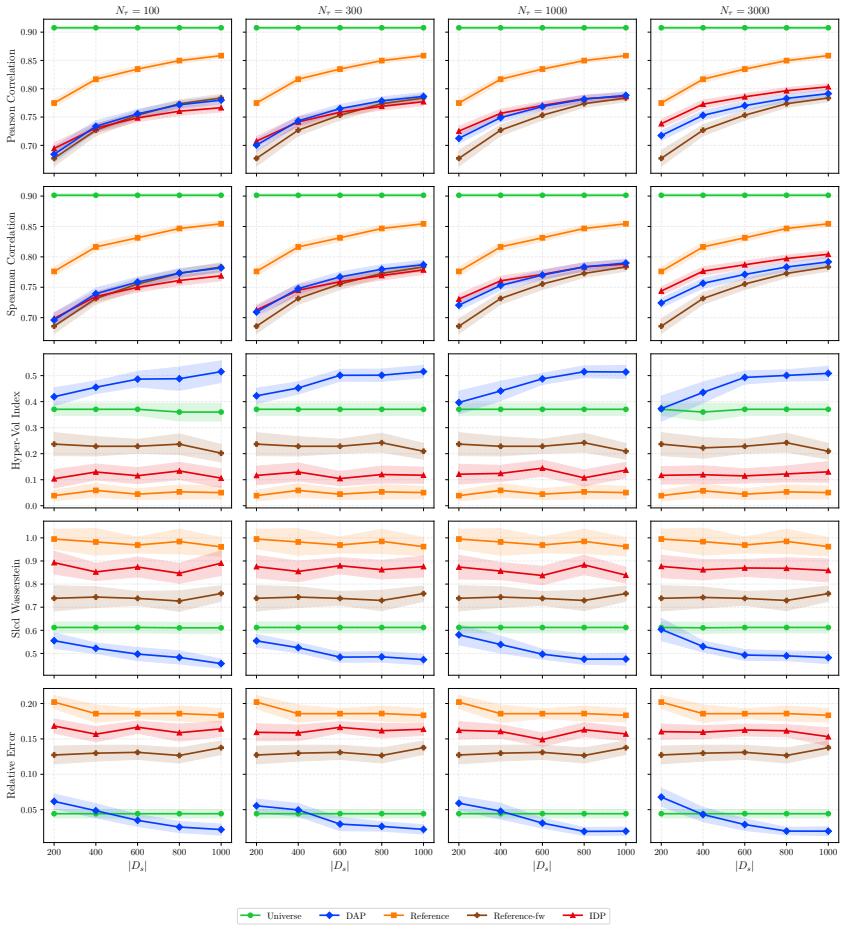


Figure 29: Prediction and steering quality as a function of the reference dataset size for the QMugs benchmark under discrete flow guidance, targeting four molecule rings. Prediction quality is measured by Spearman and Pearson correlation coefficients averaged over the time interval $[0\text{–}100]\%$. Each column corresponds to a different number of augmented trajectories.

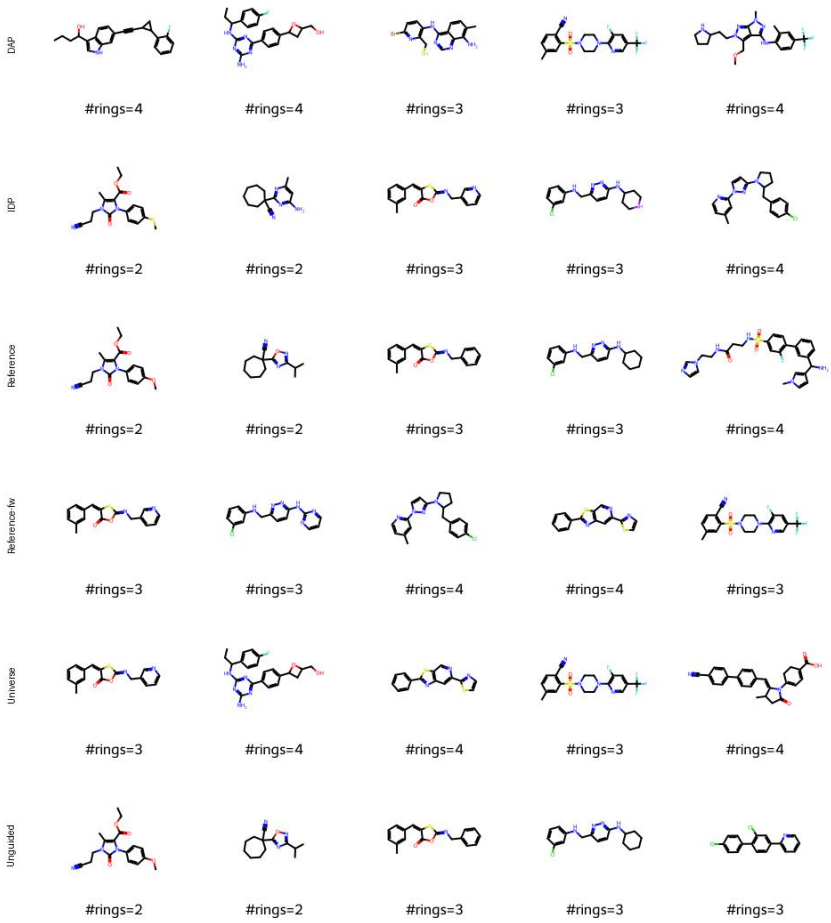


Figure 30: The first four samples generated by steering the discrete flow model with predictor guidance on the QMugs dataset ($N_\tau = 1000$, $|D_s| = 400$). The steering target is four rings. Each row corresponds to a different oracle, with *unguided* denoting prior generation without guidance.

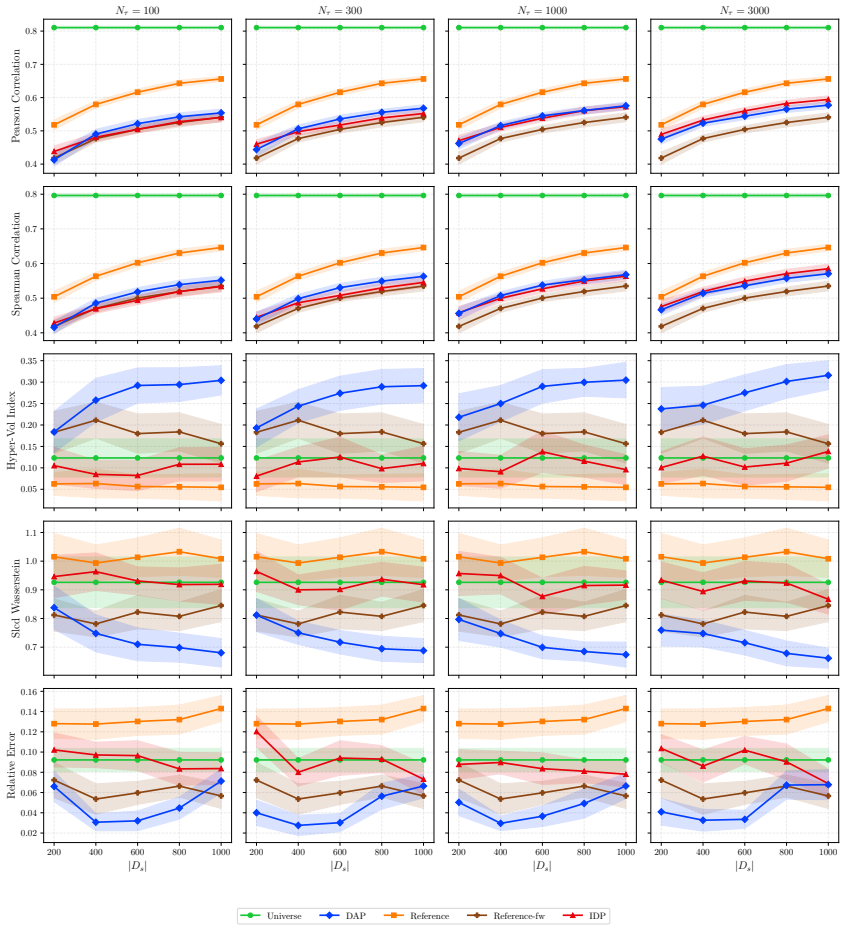


Figure 31: Prediction and steering quality as a function of the reference dataset size for the QMugs benchmark under discrete flow guidance, targeting a lipophilicity of four. Prediction quality is measured by Spearman and Pearson correlation coefficients averaged over the time interval $[0\text{–}100]\%$. Each column corresponds to a different number of augmented trajectories.

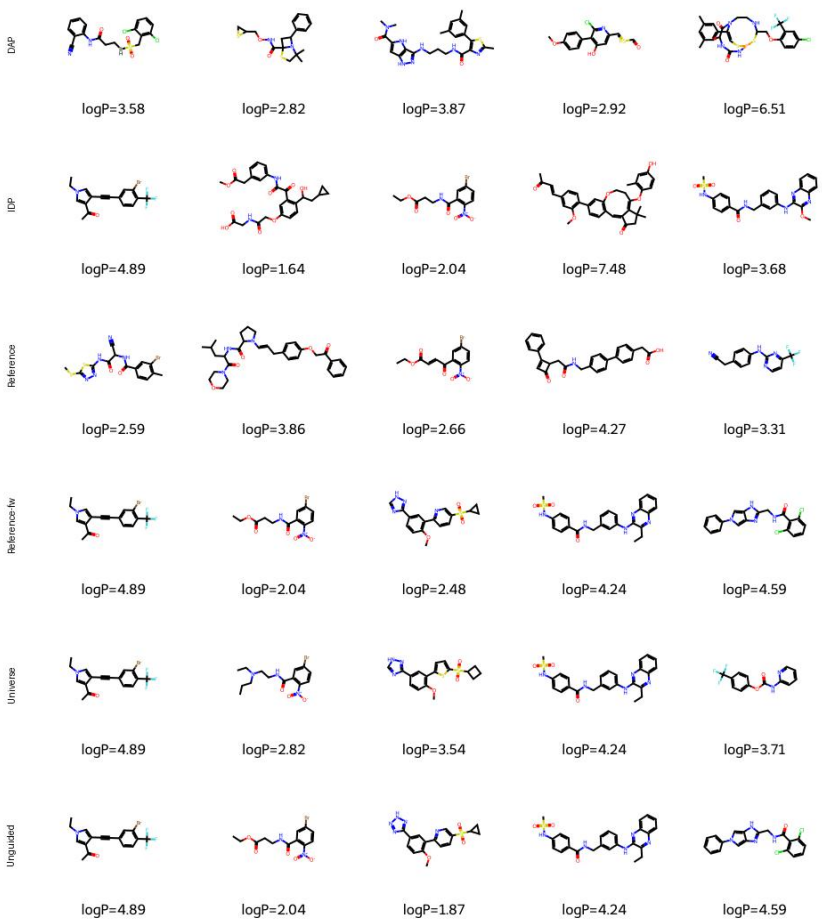


Figure 32: The first four samples generated by steering the discrete flow model with predictor guidance on the QMugs dataset ($N_\tau = 1000$, $|D_s| = 600$). The steering target is a lipophilicity of four. Each row corresponds to a different oracle, with *unguided* denoting prior generation without guidance.

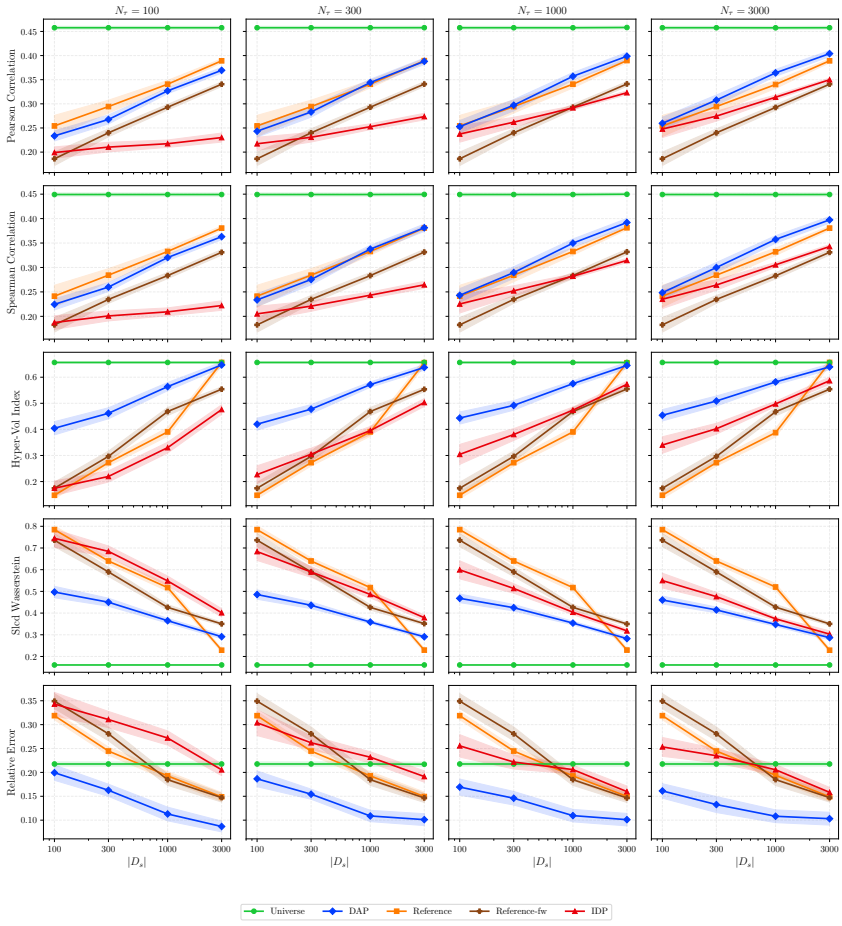


Figure 33: Prediction and steering quality as a function of the reference dataset size for the QMugs benchmark under predictor guidance in the latent space of molecules, targeting a lipophilicity of five. Prediction quality is measured by Spearman and Pearson correlation coefficients averaged over the time interval $[0\text{–}100]\%$. Each column corresponds to a different number of augmented trajectories.

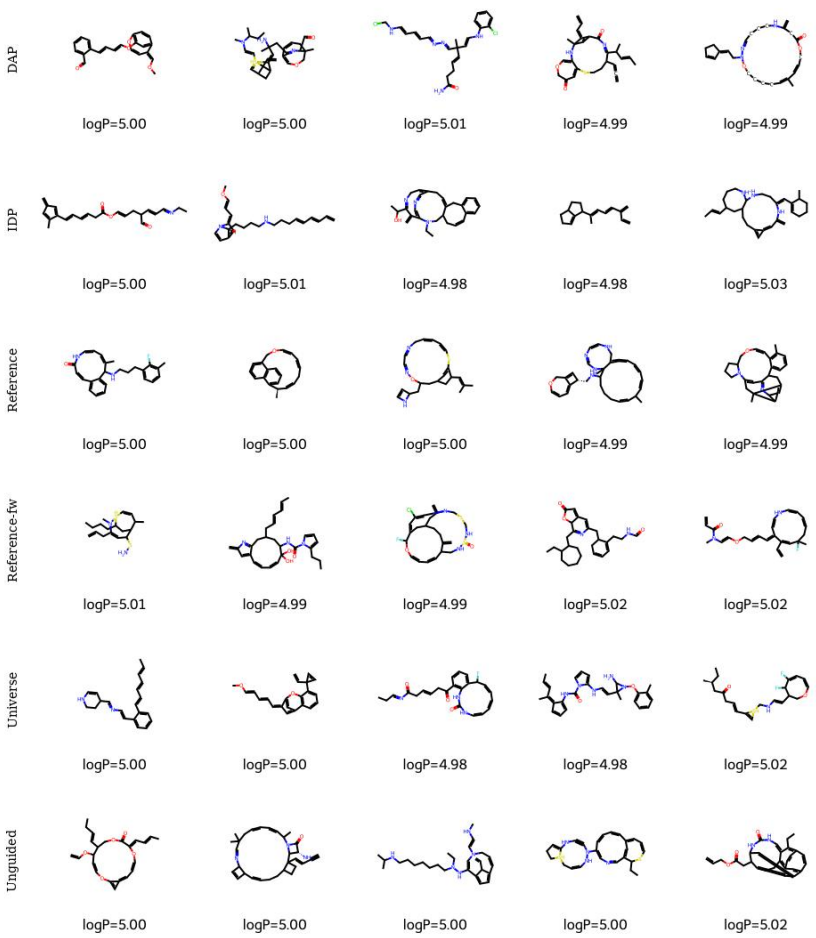


Figure 34: The top five molecules generated by predictor guidance in the latent space of molecules on the QMugs dataset, sorted by relative distance to the target lipophilicity of five ($N_\tau = 1000$, $|D_s| = 100$). Each row corresponds to a different oracle, with *unguided* denoting prior generation without guidance.