SCALABLE INFERENCE-TIME STEERING IN MOLECULAR DESIGN WITH MULTIMODAL META FLOW MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Biological design problems naturally have continuous (e.g. molecular structures), discrete (e.g. DNA residue types) and often both modalities (e.g. protein structure-sequence co-design). Diffusion and CTMC-based methods have been developed for multimodal generation, and, for cross-modal tasks, the two are typically combined. However, it remains challenging to adapt these methods at inference-time to fulfill design constraints. Recent developments of one-shot samplers such as Meta Flow Maps (MFM) have demonstrated high controllability for image generation, showcasing the power of inference-time steering with accurately estimated reward gradients. Here, we adapt Meta Flow Maps to biological domains, and develop MFM framework both in discrete (dMFM) and multimodal settings (multiMFM). We demonstrate dMFM by designing DNA sequences with a target mechanical property. MultiMFM trained on the molecular datasets QM9 and GEOM-Drugs are applied to design molecules of a given physical property. In both cases, our steering algorithms improve target attainment using one reward-independent MFM per base generator.

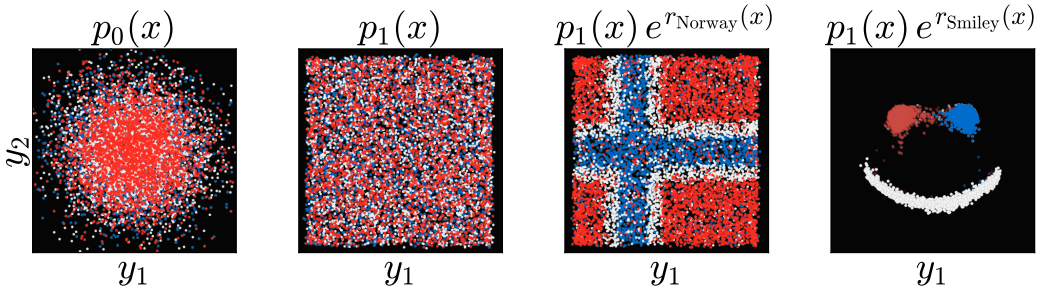


Figure 1: Examples of posterior guidance using a multimodal Meta Flow Map. $p_0(x)$ is the noise distribution, and $p_1(x)$ is the base model distribution. The Norway example steers only color coordinates, while the Smiley example steers both position and color.

1 INTRODUCTION

De novo design of chemical and biological objects increasingly relies on generative models that can represent complex molecular structures (Ingraham et al., 2023; Watson et al., 2023; Didi et al., 2026; Pacesa et al., 2025). In small-molecule design, this requires modeling discrete atom and bond types together with continuous three-dimensional geometry (Dunn and Koes, 2026). In DNA sequence design, the object is discrete, but the design objectives often depend on biological or biophysical properties that are difficult to specify directly from sequence alone (Nisonoff et al., 2025; Brookes et al., 2019; Zrimec et al., 2022). Across these settings, the central challenge is not only to generate valid samples, but to steer generation toward task-specific objectives.

The need for flexible steering at inference-time is a key motivation for controllable generative modeling. Molecular and sequence design tasks often require optimizing different combinations of objectives, such as validity, stability, binding affinity, activity, or constraint satisfaction. Recent work has therefore developed guidance and reward-driven sampling methods for discrete biological se-

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.

quences and 3D molecular generation, including predictor-guided discrete diffusion and flow models, test-time reward refinement, multi-objective sequence guidance, and property-guided structure-based drug design (Nisonoff et al., 2025; Uehara et al., 2025; Tang et al., 2025; Chen et al., 2025; Kadan et al., 2025; Hu et al., 2025; Stark et al., 2024; Jian et al., 2026; Dhariwal and Nichol, 2021). Training or fine-tuning a separate generative model for every new reward specification limits the rate of experimentation. Instead, we seek to keep a base generator fixed and steer it through reward-guided inference, so that different objective combinations can be explored without retraining the generator from scratch.

1.1 Related Work

Consistency models and one-step samplers. A separate line of work accelerates diffusion and flow-based sampling by learning maps that approximate the solution operator of the generative dynamics. Progressive distillation and consistency models learn fast one-step or few-step samplers for diffusion models (Salimans and Ho, 2022; Song et al., 2023b; Boffi et al., 2025; Zhou et al., 2025; Geng et al., 2025; Kim et al., 2024), while flow-map methods formulate this as learning the solution map associated with a probability-flow ODE. Discrete Flow Maps extend to categorical generative models (Potaptchik et al., 2026b). Meta Flow Maps extend flow-maps to stochastic posterior sampling: conditioned on an intermediate state x_t, an MFM draws samples from $p_{1|t}(\cdot \mid x_t)$, enabling Monte Carlo estimates of value functions and reward gradients (Potaptchik et al., 2026a). Diamond Maps and GLASS Flows are closely related stochastic sampling and reward-alignment methods for flow and diffusion models (Holderrieth et al., 2026a;b). Our contribution is to adapt this posterior-sampling view by deriving a discrete and multimodal analogue for DNA sequence generation and molecular design. Additional biological background and related work are in Appendix A.

In this paper, we adapt Meta Flow Maps (Potaptchik et al., 2026a) to biological design, developing inference-time steering frameworks for multimodal molecular generation (multiMFM), discrete sequence generation (dMFM). We instantiate these methods for small-molecule and DNA sequence design using domain-specific base generators. Our main contributions are:

- We construct multiMFM for small-molecule generation. MultiMFM steers towards desired chemical-property targets at high accuracy, using a single distilled MFM that can be reused across property objectives;
- We formulate discrete GLASS posterior sampling and derive discrete MFM;
- We combine continuous and discrete MFMs into the multiMFM framework for mixed discrete–continuous steering;
- dMFM enables inference-time optimization of generated DNA sequences toward prescribed predicted cyclizability values.

2 Methods

We review the mathematical framework and inference-time steering with reward gradient in the context of flow matching. Then we discuss an efficient estimator of the reward gradient $\nabla V_t(x)$ enabled by the recent development of Meta Flow Maps (Potaptchik et al. (2026a)). We develop the discrete and multimodal analogy of Meta Flow Maps (dMFM and multiMFM) which are applied for multimodal inference-time steering.

Inference-time Steering. Let a flow matching model $b_t(x_t)$ sample a distribution $p_1(x_1)$. In practice for a given reward function $r(x_1)$, we may need to sample the reward tilted distribution $\propto p_1(x_1)e^{r(x_1)}$. For the Gaussian-source linear interpolant (18), exact drift, score, and posterior value function give the following ODE *or* SDE sampler:

$$\text{ODE Sampler:} \quad dX_t = \left(b_t(X_t) + \frac{\sigma_t^2}{2}\nabla V_t(X_t)\right) dt; \tag{1}$$

$$\text{SDE Sampler:} \quad dX_t = \left(b_t(X_t) + \frac{\sigma_t^2}{2}\nabla \log p_t(X_t) + \sigma_t^2\, \nabla V_t(X_t)\right) dt + \sigma_t\, dB_t. \tag{2}$$

The matching schedule is $\sigma_t^2 = 2(1-t)/t$ for $0 < t < 1$. The exact-law statement concerns the ideal dynamics; learned posterior maps, finite Monte Carlo budgets, numerical integration, and practical guidance modifications introduce approximation error. The value function $V_t(x_t)$ is defined as $V_t(x_t) = \log \mathbb{E}_{X_1 \sim p(\cdot|x_t)}[e^{r(X_1)} \mid X_t = x_t]$. Various methods have been proposed to estimate $V_t(x_t)$. One estimator of $V_t(x_t)$ uses Meta Flow Maps $\Phi(\,\cdot\,;\, t,\, x_t)$(Potaptchik et al. (2026a)); Diamond Maps provide a related posterior-sampling construction (Holderrieth et al. (2026a)). Meta Flow Maps is an efficient one-step sampler that send Gaussian noise ϵ to $\Phi(\epsilon; t, x_t)$ a sample of the posterior distribution $p_1(\cdot|x_t)$. Drawing N noises ϵ_i and evaluate the meta flow map on ϵ_i for i from 1 to N gives us N samples of $p_1(\cdot\,|x_t)$. We can form a Monte Carlo estimator of the gradient of the value function:

$$\widehat{\nabla V}_t^r(x_t) = \nabla_x \log \left(\frac{1}{N} \sum_{i=1}^{N} \exp(r(\Phi(\epsilon_i\,; t, x_t))) \right) \tag{3}$$

for reward $r(\cdot)$ differentiable in its arguments, provided differentiation and expectation can be interchanged. This finite-N log-mean estimator is generally biased. For discrete outputs, differentiating through simplex-valued predictions estimates a relaxed surrogate; it does not by itself justify pathwise differentiation through exact categorical samples. A gradient-free alternative is

$$\frac{\sigma_t^2}{2} \widehat{\nabla V}_t^r(x_t) = \left(\dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t} \beta_t \right) \sum_{i=1}^{N} \mathrm{softmax}_i(r(\Phi(\epsilon_i;\, t, x_t))) \cdot \Phi(\epsilon_i;\, t, x_t) + \frac{\dot{\alpha}_t}{\alpha_t} x_t - b_t(x_t). \tag{4}$$

Here $X_t = \alpha_t X_0 + \beta_t X_1$, with $\alpha_t = 1 - t$ and $\beta_t = t$ for the linear interpolant.

Meta flow maps is distilled from the base model $b_t(x_t)$. We illustrate inference-time steering with Meta Flow Map in Figure 2:

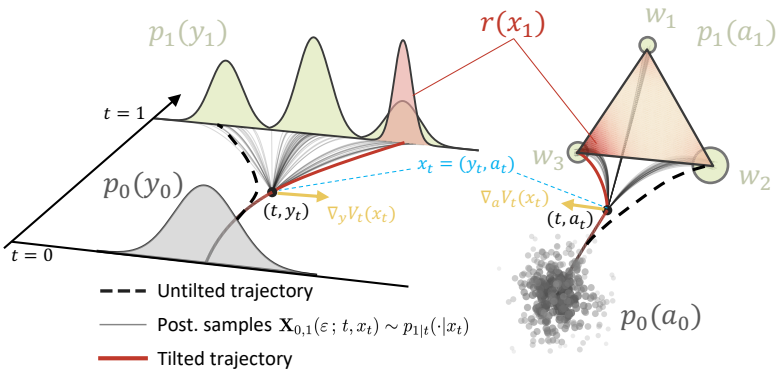


Figure 2: **inference-time steering using multimodal Meta Flow Map.** Unguided trajectory is in dotted black. Conditional on current state (t, y_t, a_t), multiMFM $\mathbf{X}_{0,1}(\epsilon_y, \epsilon_a\,; t, y_t, a_t)$ obtains look-aheads from the joint posterior $p_{1|t}(y_1, a_1 \mid y_t, a_t)$, as shown in light gray. Reward $r(x_1)$ evaluates the look-aheads, sending gradient signal back to guide path (t, y_t, a_t).

Discrete Meta Flow Maps. Here we develop the general framework adapting Meta Flow Maps to the discrete domain. We follow the notation of Potaptchik et al. (2026b) and work with K discrete tokens. In dMFM for DNA, we have $K = 4$. Let $\mathcal{V} = \{e_1, \ldots, e_K\} \subset \mathbb{R}^K$ be the K vertices of Δ^{K-1}, the probability simplex in $\mathbb{R}^K$. A vertex corresponds to a token.

Discrete flow matching learns a denoiser $\psi(x_t) = \mathbb{E}[X_1|X_t = x_t]$ as the expected endpoint when the flow passes through x_t instead of the velocity $b_t(x_t)$. Since X_1 is supported on $\mathcal{V}$, we have $\psi_t(x_t) \in \Delta^{K-1}$. If the interpolant between noise and data is linear as in equation (18), then the velocity $b_t(x_t)$ is analytic in $\psi(x_t)$:

$$b_t(x) = \frac{\psi_t(x) - x}{1 - t}. \tag{5}$$

$b_t(x_t)$ transports from Gaussian noise $p_0(x_0)$ to a categorical distribution $p_1(x_1)$ supported on vertices $\mathcal{V}$. Given $b_t(x_t)$ and intermediate state x_t, we extend the framework of GLASS (Holderrieth

et al. (2026b)) to sample from $p(x_1|X_t = x_t)$ by an inner flow matching process. The velocity field $\bar{b}_s(\bar{x}_s\,;\,t, x_t)$ of the inner flow matching process is analytic in b_t as follows

$$\bar{b}_s(\bar{x}_s; t, x_t) = w_1\,\bar{x}_s + w_2\,\psi_{t^*}(S(\bar{x}_s, x_t)) + w_3\,x_t, \tag{6}$$

where w_1, w_2, w_3 are GLASS weights, and t^* is a reparametrization of t and s. The sufficient statistic $S(\bar{x}_s,\, x_t)$ is an analytic function linear in $\bar{x}_s$ and x_t. Drawing $\bar{x}_0 \sim \mathcal{N}(0, \mathbb{I})$ and solve for $\bar{x}_s$ from the ODE:

$$\frac{d}{ds}\bar{x}_s = \bar{b}_s(\bar{x}_s; t, x_t) \tag{7}$$

from $s = 0$ to $s = 1$ yields a sample of $p(x_1|X_t = x_t)$. The inner velocity field $\bar{b}_s(\bar{x}_s;\, t, x_t)$ can be distilled into a Discrete Meta Flow Map (dMFM) $\mathbf{X}_{0,1}(\varepsilon\,;\,t, x_t)$ to draw a sample from $\mathcal{V}$ according to $p(x_1|X_t = x_t)$ in one shot.

Definition 1. *A Discrete Meta Flow Map is a mapping $\mathbf{X}_{s,u}(\cdot\,;\,t, x_t) : \mathbb{R}^K \to \mathbb{R}^K$ is the solution operator for the ODE* (7)*. That is for every $t < 1$ and x_t and $0 \le s \le u < 1$, we have:*

$$\mathbf{X}_{s,u}(\bar{x}_s\,;\,t, x_t) = \bar{x}_u, \tag{8}$$

for $\bar{x}_s$ following the ODE (7)*. The terminal map at $u = 1$ is understood as an almost-sure endpoint limit.*

We can reparameterize dMFM in terms of the *mean meta denoiser* $\mathbf{\Psi}_{s,u}$:

$$\mathbf{X}_{s,u}(\bar{x}_s\,;\,t, x_t) = \frac{1-u}{1-s}\bar{x}_s + \frac{u-s}{1-s}\mathbf{\Psi}_{s,u}(\bar{x}_s\,;\,t, x_t), \tag{9}$$

as Figure 2 illustrates. The meta denoiser $\mathbf{\Psi}_{s,u}$ is the endpoint of the secant line between $\bar{x}$ and $\mathbf{X}_{s,u}(\bar{x}_s\,;\,t, x_t)$ on the simplex Δ^{K-1}. In training, we ensure $\mathbf{\Psi}_{s,u}$ to lie on Δ^{K-1} by parameterizing it with logits $\hat{\mathbf{z}}_{s,u}$:

$$\hat{\mathbf{\Psi}}_{s,u}(\bar{x}_s\,;\,t, x_t) = \mathrm{softmax}\big(\,\hat{\mathbf{z}}_{s,u}(\bar{x}_s\,;\,t, x_t)\big) \in \Delta^{K-1}. \tag{10}$$

The neural network represents $\hat{\mathbf{z}}_{s,u}(\bar{x}_s\,;\,t, x_t)$. The meta denoiser $\hat{\mathbf{\Psi}}_{s,u}$ is related to the *meta velocity* $\hat{\mathbf{v}}_{s,u}$ analogous to equation (5):

$$\hat{\mathbf{v}}_{s,u}(\bar{x}_s; t, x_t) = \frac{\hat{\mathbf{\Psi}}_{s,u}(\bar{x}_s\,;\,t, x_t) - \bar{x}_s}{1-s}. \tag{11}$$

Training dMFM. To train dMFM, we first notice the following fact:

Proposition 1 (**Diagonal Meta Denoiser**). *For the linear interpolant* (18)*, the meta denoiser satisfies*

$$\mathbf{\Psi}_{s,s}(\bar{x}_s\,;\,t, x_t) = \psi_{t^*}(S(\bar{x}_s, x_t)), \tag{12}$$

where ψ is the denoiser of the base discrete flow matching model. The base denoiser is evaluated at t^* which is the GLASS reparameterization of t and s. $S(\bar{x}_s, x_t)$ is the GLASS sufficient statistics.

See Appendix C.2 for a proof and analytic formulae of t^* and $S(\bar{x}_s, x_t)$. To enforce Proposition 1, we introduce the diagonal loss as a KL divergence (equivalent to cross entropy up to the fixed teacher entropy) between $\hat{\mathbf{\Psi}}_{s,s}$ and ψ_{t^*}:

$$\mathcal{L}_{\mathrm{diag}}(\hat{\mathbf{\Psi}}) = \mathbb{E}\left[\, D_{KL}\Big(\psi_{t^*}(S(\bar{x}_s, x_t))\Big\|\hat{\mathbf{\Psi}}_{s,s}(\bar{x}_s\,;\,t, x_t)\Big)\,\right]. \tag{13}$$

The solution operator also satisfies consistency identities; the Eulerian, progressive, and Lagrangian formulations are given in Appendix C.2. Each of the three losses $\mathcal{L}_{\mathrm{ESD}}$ $\mathcal{L}_{\mathrm{PSD}}$, and $\mathcal{L}_{\mathrm{LSD}}$ is called a consistency loss $\mathcal{L}_{\mathrm{con}}$. The overall training loss of $\hat{\mathbf{\Psi}}$ is the sum of the diagonal loss (13) with any of the consistency loss

$$\mathcal{L}(\hat{\mathbf{\Psi}}) = \mathcal{L}_{\mathrm{diag}}(\hat{\mathbf{\Psi}}) + \mathcal{L}_{\mathrm{con}}(\hat{\mathbf{\Psi}}). \tag{14}$$

We use ESD for consistency training; PSD and LSD are theoretical alternatives.

Multimodal Steering. Suppose state x_t has a continuous modality y_t and discrete modality a_t, i.e. $x_t = (y_t, a_t)$ as in Figure 2. A multimodal Meta Flow Map (multiMFM) samples the joint $x_1 = (y_1, a_1)$ from the posterior $p_{1|t}(y_1, a_1 \mid y_t, a_t)$ and has continuous and discrete output components:

$$\mathbf{X}_{s,u}(\bar{y}_s, \bar{a}_s\,;\, t, y_t, a_t) = (\Phi^c_{s,u},\, \mathbf{X}^d_{s,u})(\bar{y}_s, \bar{a}_s\,;\, t, y_t, a_t) \tag{15}$$

where Φ^c is the continuous MFM as in Potaptchik et al. (2026a) and $\mathbf{X}^d$ is the dMFM introduced before. Both components depend on the joint inner and outer states; no independence between modalities is assumed. The value function $V(x_t) = V(y_t, a_t)$ can depend on both modalities. At inference-time, the paths of y_t and a_t are steered with partial gradients of $V(y_t, a_t)$ against y_t and a_t respectively.

$$\begin{cases} dY_t = (b^y_t(Y_t, A_t) + \frac{\sigma^2_{y,t}}{2}\, \partial_y V_t(Y_t, A_t))\, dt \\ dA_t = (b^a_t(Y_t, A_t) + \frac{\sigma^2_{a,t}}{2}\, \partial_a V_t(Y_t, A_t))\, dt, \end{cases} \tag{16}$$

where b^y_t is the velocity field of modality y and b^a_t is discrete flow matching velocity in the sense of equation (5). For the common linear schedule, $\sigma^2_{y,t} = \sigma^2_{a,t} = 2(1-t)/t$. These ODEs are used for the reported steering experiments and follow from (1) by setting $X_t = (Y_t, A_t)$ and split the gradient into $\nabla = (\partial_y, \partial_a)$. The partial gradients are estimated as in equation (3), with the same relaxation caveat for categorical outputs. Independent modality schedules require a matching joint posterior; merely changing the two guidance coefficients does not preserve the above exact-law argument.

3 Experiments

We evaluate the dMFM in DNA property guidance. We evaluate the multimodal MFM on a molecule system containing discrete (atom types) and continuous (coordinates) modalities. In Section 3.1, we train an unconditional DNA sequence generator and distill it into a dMFM. At inference time, we use reward guidance to generate DNA sequences with predicted cyclizability scores close to specified targets. In Section 3.2, we perform property guidance on QM9 Ramakrishnan et al. (2014) and GEOM-DRUGS Axelrod and Gómez-Bombarelli (2022). We further compare dMFM against common inference-time steering frameworks and examine posterior diversity and Monte Carlo error across sequence lengths. A synthetic multimodal illustration is shown in Figure 1.

Multimodal flag example. We steer colored particles in $\mathbb{R}^2$ into a Norway flag or a smiley (Figure 1). For the flag, we steer only color, preserving the square spatial distribution. For the smiley, we steer position and color together. Details are in Appendix F.7.

3.1 Mechanical Property-Guided DNA Design

We train a dMFM DNA sequence model and evaluate reward-guided sampling. First, we train a base discrete flow model in the ambient sequence space $(\mathbb{R}^4)^L$, with each nucleotide represented by a vertex of $\Delta^3 \subset \mathbb{R}^4$. The model architecture is similar to that of Potaptchik et al. (2026b). After training the base model, we distill it into a dMFM using the Eulerian self-distillation (ESD) loss. As done in Potaptchik et al. (2026a), we condition the dMFM network on the outer time and state (t, x). While the dMFM works as a one-shot sampler, empirically we find using 2-4 steps gives increased performance. Architectural details are given in Appendix E.1.

Sequence-encoded DNA mechanics can carry regulatory information beyond transcription-factor motif content. Loop-seq measurements showed that intrinsic cyclizability is sequence dependent and that promoter-proximal yeast nucleosome-depleted regions contain unusually low-cyclizability DNA (Basu et al., 2021). Subsequent work used sequence-based cyclizability prediction to study mechanical selection across eukaryotic genomes (Park et al., 2024), while related studies linked local nucleosomal DNA mechanics to transcription-factor binding (Mariani et al., 2025). We evaluate dMFM for DNA design guided by predicted cyclizability. Given a target score, a pretrained guide predictor defines a terminal reward favoring 50-bp sequences whose predicted scores are close to the target. Reward-guided sampling generates candidate sequences whose mechanical properties and any regulatory effects require experimental validation.

Table 1: Independent evaluation of cyclizability-guided DNA generation. The guide and oracle regressors are trained on disjoint parent sequences. Only the guide is used during sampling. Brackets denote 95% confidence intervals. Positive paired improvement means that guidance reduces oracle target error.

Target $y^\star$	Mean oracle score Unguided	Guided	Unguided MAE	Guided MAE mean [95% CI]	Paired improvement
$+1$	-0.169	0.912	1.170 [1.038, 1.291]	0.157 [0.119, 0.196]	1.013 [0.881, 1.142]
-1	-0.169	-0.916	0.831 [0.709, 0.964]	0.153 [0.118, 0.190]	0.678 [0.554, 0.816]

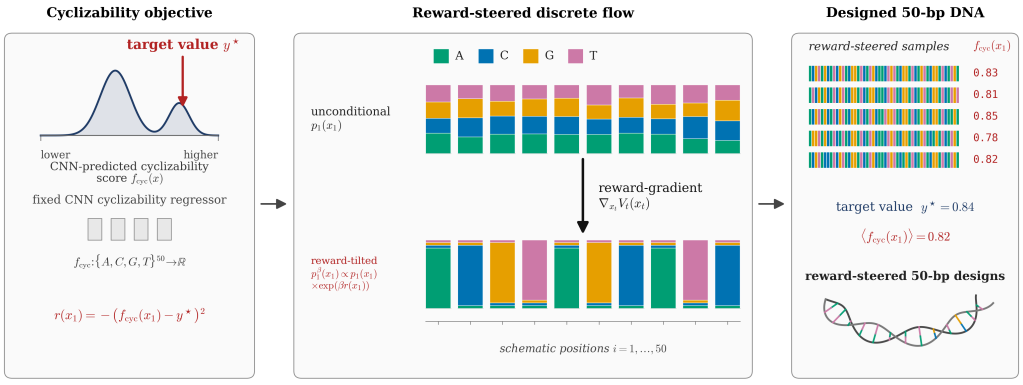


Figure 3: **Cyclizability-guided DNA design.** A guide predictor f_{guide} defines the terminal reward equation (17) and supplies gradients for sampling. Generated sequences are evaluated using a parent-disjoint oracle predictor f_{oracle}, which is not shown above and is not used during guidance. On the 58 held-out test parents (8,294 windows), the guide obtains Pearson r = 0.88 with MAE 0.159 and the oracle r = 0.87 with MAE 0.162. Their independence is in training data (disjoint parents, different seeds), not architecture.

We evaluate cyclizability-guided DNA design for target scores $y^\star \in \{-1, +1\}$. To separate guidance from evaluation, we train two cyclizability regressors, a guide f_{guide} and an oracle f_{oracle}, on disjoint sets of parent sequences. Only f_{guide} is differentiated during sampling, using the terminal reward

$$r_{\mathrm{guide}}(x_1; y^\star) = -\left(f_{\mathrm{guide}}(x_1) - y^\star\right)^2 . \tag{17}$$

The oracle is for evaluation and never used for guidance. For each target, we generate 1,000 guided–unguided pairs of 50-bp sequences, using identical initial noise within each pair. We report oracle-evaluated target error in Table 1. This protocol tests whether target attainment transfers to a predictor trained on parent-disjoint data, rather than measuring success only with the model being optimized. Target attainment transfers strongly to the parent-disjoint oracle. Guidance reduces oracle MAE from 1.17 to 0.16 for $y^\star = +1$ and from 0.83 to 0.15 for $y^\star = -1$. This does not establish experimental biological validity, because the oracle remains a learned predictor, but it shows that the observed gains are not solely an artifact of evaluating with the regressor used for guidance. The guided errors of 0.16 and 0.15 are comparable to the oracle's own MAE of 0.16 against measured C0 on the held-out parents. The experiment therefore establishes that guidance reaches the target region, not that it attains the target to within 0.15, resolving finer differences would require a more accurate evaluator than either regressor.

To test steering without any learned reward evaluator, we additionally use two exactly computable sequence rewards: a smooth occurrence score for the motif `TTTTTC`, and a conjunction requiring both `TTTTTC` and `AAAATT`. Across lengths $L \in \{50, 100, 200, 400\}$, guidance improves around 95% of matched guided-unguided pairs. The raw loop-seq assay provides 100-bp records, so a central 50-bp yeast window flanked by fixed 25-bp adapters, with a measured C0 value. Because consecutive central windows within a region overlap by 43 bp, stride 7, they reconstruct into 576 contiguous 1,044-bp parent sequences. Parents are split once into disjoint sets. Appendix F.1.

The separate diversity analysis shows more local clustering for the negative target despite complete sample uniqueness (Appendix B). Additional quantitative results for this DNA design task, including predicted cyclizability score distributions for each target value, are reported in Appendix F.4.

3.2 Multimodal Property Guidance on TABASCO

To demonstrate the capabilities of multiMFM on real data, we trained a molecule model where the discrete types are atoms and the continuous types are coordinates. The architecture is inspired by Tabasco Vonessen et al. (2026), but where we have modified their CTMC for discrete types with a discrete flow matching with a continuous gaussian prior flowing to the simplex. After training this model and verify strong PB valid results (we here ignore Hydrogen atoms and only look at heavy atoms, as they also do in Tabasco and TFG-flow), we train a multimodal Meta Flow Map (multiMFM) to sample posterior samples. We perform property guidance using the heavy-atom setup of TFG-Flow. We import its published baseline results, evaluated using RDKit validity (TFG-Flow in turn reports the Upper bound, #Atoms, Lower bound, Cond-EDM and EEGSDE rows from Bao et al. (2023), and the training-free diffusion rows from Ye et al. (2024)), while our reported errors use the stricter PoseBusters (PB) validity filter. The resulting evaluation populations differ. Note that while in Table 1 in the TFG-flow they distinguish between training based and training free methods. The "training based" category includes methods that learn property-conditioned generation or guidance components. However, we only train the multiMFM once, and then any reward function can be applied. Furthermore, training the multiMFM is only for increased sampling efficiency: we could also have a "training-free" approach by simply performing posterior guidance with the multimodal GLASS-flow.

Table 2: Mean absolute error (MAE) for molecular property guidance. All published baseline values are reproduced from Table 1 of TFG-Flow (Lin et al., 2025). Energy-related properties are reported in meV. Our rows report MAE over PB-valid molecules. Lower is better.

Method	**Model category**	C_v	μ	α	$\Delta\varepsilon$	$\varepsilon_{\text{HOMO}}$	$\varepsilon_{\text{LUMO}}$
Upper bound	Reference	6.87	1.61	8.98	1464	645	1457
#Atoms	Reference	1.97	1.05	3.86	886	426	813
Lower bound	Reference	0.040	0.043	0.09	65	39	36
Cond-EDM (Hoogeboom et al., 2022)	Continuous Diffusion	1.065	1.123	2.78	671	371	601
EEGSDE (Bao et al., 2023)	Continuous Diffusion	0.941	0.777	2.50	487	302	447
UniGEM (Feng et al., 2025)	Continuous Diffusion	0.873	0.805	2.22	511	**233**	592
Cond-Flow (Campbell et al., 2024)	Multimodal Flow	1.52	0.962	3.10	805	435	693
DPS (Chung et al., 2023)	Continuous Diffusion	5.26	63.2	51169	1380	744	NA
LGD (Song et al., 2023a)	Continuous Diffusion	3.77	1.51	7.15	1190	664	1200
FreeDoM (Yu et al., 2023)	Continuous Diffusion	2.84	1.35	5.92	1170	623	1160
MPGD (He et al., 2024)	Continuous Diffusion	2.86	1.51	4.26	1070	554	1060
UGD (Bansal et al., 2024)	Continuous Diffusion	3.02	1.56	5.45	1150	582	1270
TFG (Ye et al., 2024)	Continuous Diffusion	2.77	1.33	3.90	893	568	984
TFG-Flow (Lin et al., 2025)	Multimodal Flow	1.75	0.817	2.32	804	364	941
multiMFM (Ours)	Multimodal Flow	**0.42**	0.84	**1.12**	600	262	**392**
multiMFM-SS (Ours)	Multimodal Flow	**0.40**	**0.69**	**1.10**	**462**	252	**316**

MultiMFM errors are lower than the published TFG-Flow values on most properties. Our mean PB-validity is 91.4%, compared with the base model's 98.7%. On top of multiMFM, we introduce a steer-search trick, and present multiMFM-SS. Steer-search integrates the same guided ODE as multiMFM, but stores on CPU the look-aheads that multiMFM computed. At the end of integration, multiMFM-SS returns the PB-valid molecule with lowest guide-predicted error among the look-aheads and endpoint, falling back to the endpoint if none is PB-valid. Steer-search reuses the existing posterior samples without additional generative-model evaluations, but adds molecule conversion, validity checking, selection, and storage costs. It further improves the reported MAEs as in Table 2 and gives mean PB validity of 97.1%. Refer to table 21 for details about PoseBuster Validity. QM9 uses four inner steps and 32 posterior samples per outer step; GEOM-Drugs uses 10 flow-map inner steps and four posterior samples (Appendix F.8).

We scale multiMFM to GEOM-DRUGS containing molecules up to 71 heavy atoms, compared with 9 for QM9 (Table 19). We follow the benchmark of TFG-Flow Lin et al. (2025) on the structural

fingerprint task. Let the target fingerprint be c, and a guide predictor $\mathcal{E}$ predict the fingerprint from a generated molecule G. The positive tilt weight is $R(G) = \exp(-||\mathcal{E}(G) - c||^2)$, corresponding to the log-reward $r(G) = \log R(G) = -||\mathcal{E}(G) - c||^2$ in (3), as in TFG-Flow. Our $\mathcal{E}(G)$ is off-the-shelf from TFG-Flow. To evaluate independent of the guide predictor, we steer with $\mathcal{E}(G)$ but for the output molecule G' we count the fingerprints directly without any oracle and report the Tanimoto similarity with the target c. Our similarity improves on the baseline TFG-Flow, although still moderate.

Table 3: GEOM-DRUGS structural fingerprint guidance. The two reference values are from TFG-Flow. multiMFM yielded 87.5% PB-valid structures, compared with 92.0% for its base generator.

Method	Tanimoto similarity ↑
Unconditional generation (TFG-Flow "Upper bound")	0.170 ± 0.001
TFG-Flow (Lin et al., 2025)	0.208 ± 0.002
multiMFM	**0.227**

3.3 Scaling and Robustness Study

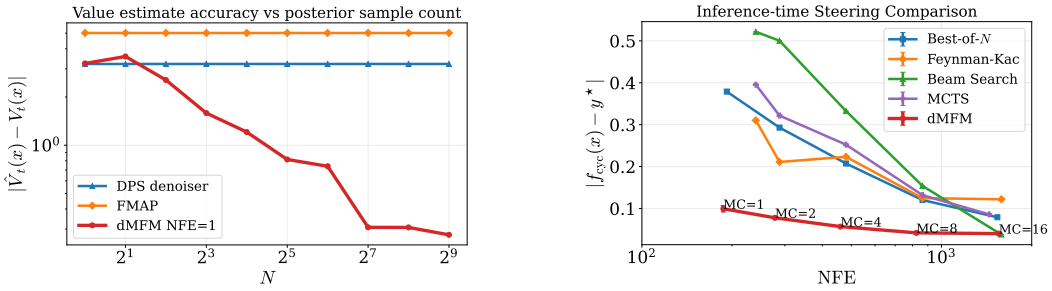


Figure 4: Left: Value-function estimation error as the number of Monte Carlo samples N increases. Right: Target error versus reported network function evaluations (NFEs).

Posterior Recovery. As a first benchmark, we assess its ability to estimate the value function $V_t(x) = \log \mathbb{E}[e^{r(X_1)} \mid X_t = x]$ through with N estimated samples from the posterior distribution $p_{1|t}(\cdot|x)$. We compare against the denoiser approximation $V_t(x) \approx r(\mathbb{E}[X_1 \mid X_t = x])$ given both by DPS (Chung et al. (2023)) and a one-step Flow Map (FMAP). For the FMAP, we estimate a best case accuracy by integrating along its flow matching $b_t(x)$ trajectory. We observe strong scaling of the dMFM with number of Monte Carlo samples N.

Steering. We benchmark dMFM against well-known inference-time steering methods as a function of neural function evaluations (NFEs). We compare against Best-of-N, Feynman–Kac steering, beam search, and Monte Carlo tree search (MCTS), as described in Didi et al. (2026). We define the reward as $r(x) = -(f_{\text{cyc}}(x) - y^\star)^2$, where $f_{\text{cyc}}(x)$ is the cyclizability predictor and $y^\star$ is the target cyclizability value, as in Section 3.1. dMFM achieves low target error at low NFEs, the regime most relevant for practical sampling. The reported NFE counts do not assume parallel execution. NFEs alone do not establish equal wall-clock cost, especially when comparing forward sampling with guidance that requires backpropagation. Posterior pairwise Hamming distances remain close to the GLASS reference from 50 to 400 bp (Appendix F.5).

The gradient diagnostics use three rewards: a quadratic GC-content target, a smooth occurrence score for `TTTTTC`, and a smooth conjunction of `TTTTTC` and `AAAATT`. The raw dMFM-versus-GLASS comparison is reported separately in Appendix F.6.

Scale-free Monte Carlo convergence. Raw per-coordinate MAE is confounded by reward-gradient scale. When a sequence-level gradient is distributed over more coordinates, both its per-coordinate magnitude and its absolute estimation error can decrease. We thus evaluate finite-sample Monte Carlo convergence using pooled relative ℓ_1 error and cosine similarity. Figure 5 compares finite-N MC samples from dMFM evaluation with an independent GLASS reference using 2,048 samples, over 32

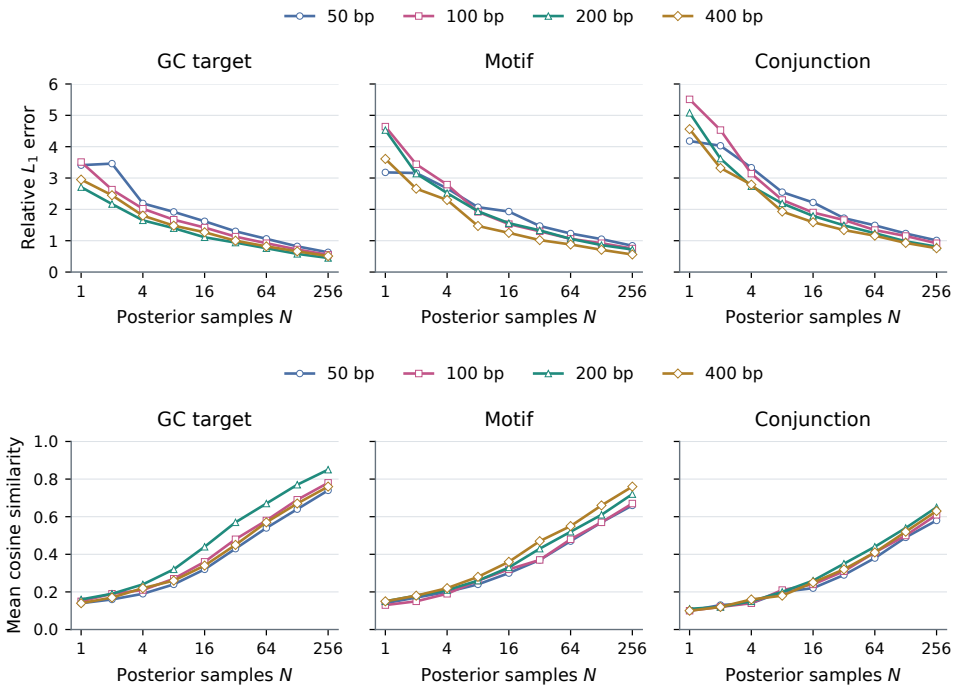


Figure 5: Finite-sample dMFM gradient estimation from 50 to 400 bp. Top: pooled relative ℓ_1 error. Bottom: mean cosine similarity. Each estimate is compared with an independent GLASS-2048 reference.

held-out conditioning states and eight noise pools per state at each length. Refer to Appendix F.3 for further metrics.

Both metrics improve as the number of posterior samples increases. Crucially, neither relative error nor directional agreement deteriorates systematically between 50 and 400 bp. We find no evidence that the parallel Monte Carlo budget must increase exponentially with sequence length. This is not a claim that a fixed budget suffices for arbitrary sequence lengths. We found $N = 8$ is an effective end-to-end operating point for DNA experiments. At $N = 8$, relative errors are 1.40–2.55 and cosine similarities are 0.18–0.32. Larger parallel budgets provide higher fidelity.

4 Discussion

We extended Meta Flow Maps to discrete (dMFM) and multimodal (multiMFM) settings. MultiMFM designs molecules with high accuracy for a given physical property value. Although QM9 guidance reduces mean PoseBusters validity from 98.7% to 91.4%, while steer-search recovers it to 97.1%. The reported property errors improve on several published reference values. On top of MultiMFM, we introduce a steer-search trick with no additional generative-model evaluations. Steer-search stores the lookaheads that MultMFM obtains along the integration trajectory and returns the best PB-valid candidate under the guide predictor, falling back to the integration endpoint if none is valid. With steer-search multiMFM improves the accuracy of molecular generation further as in Table 2. We reuse one MFM across different property rewards at inference-time instead of training separate generators for each property. We derive the dMFM as one-shot posterior sampler in the discrete domain, and showcase its scientific potential through a DNA design application. We also benchmark against other inference-time methods. Our work opens up avenues for future research on applying dMFM and multiMFM steering methods on larger scale protein and DNA structure-sequence codesign tasks.

IMPACT STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

AI USE STATEMENT

Generative AI was used to review manuscript consistency, check mathematical formulations, and to make a draft of the writeup, including tables, and implementing methods in code, clean dataset, review relevant literature. The idea and formalizing of the method, its mathematical derivation, and the experiments were all human designed. The authors are responsible for verifying all claims, experiments, and final text.

ACKNOWLEDGMENTS

Acknowledgments removed for anonymous submission.

REFERENCES

Simon Axelrod and Rafael Gómez-Bombarelli. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022. doi: 10.1038/s41597-022-01288-4.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/dfbb3d1807b21dadee735eb75069ada4-Abstract-Conference.html.

Fan Bao, Min Zhao, Zhongkai Hao, Peiyao Li, Chongxuan Li, and Jun Zhu. Equivariant energy-guided SDE for inverse molecular design. In *International Conference on Learning Representations (ICLR 2023)*, 2023. URL https://openreview.net/forum?id=r0otLtOwYW.

Aakash Basu, Dmitriy G. Bobrovnikov, Zan Qureshi, Tunc Kayikcioglu, Thuy T. M. Ngo, Anand Ranjan, Sebastian Eustermann, Basilio Cieza, Michael T. Morgan, Miroslav Hejna, H. Tomas Rube, Karl-Peter Hopfner, Cynthia Wolberger, Jun S. Song, and Taekjip Ha. Measuring DNA mechanics on the genome scale. *Nature*, 589(7842):462–467, 2021. doi: 10.1038/s41586-020-03052-3.

Aakash Basu, Dmitriy G Bobrovnikov, Basilio Cieza, Juan Pablo Arcon, Zan Qureshi, Modesto Orozco, and Taekjip Ha. Deciphering the mechanical code of the genome and epigenome. *Nature Structural & Molecular Biology*, 29(12):1178–1187, 2022. doi: 10.1038/s41594-022-00877-6.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to Build a Consistency Model: Learning Flow Maps via Self-Distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 37559–37595, 2025. doi: 10.52202/085713-1121. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2ff3605c9e442d55156d99fe4f5f228d-Abstract-Conference.html.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A. Gonzalez, Samuel H. King, David B. Li, Aditi T. Merchant, Eric Nguyen, Chiara Ricci-Tam, David W. Romero, Jonathan C. Schmok, Ali Taghibakhshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K. Wang, Michael T. Pearce, Elana Simon, Etowah Adams, Zachary J. Amador, Euan A. Ashley, Stephen A. Baccus, Haoyu Dai, Steven Dillmann, Stefano Ermon, Daniel Guo, Michael H. Herschl, Rajesh Ilango, Ken Janik, Amy X. Lu, Reshma Mehta, Mohammad R. K. Mofrad, Madelena Y. Ng, Jaspreet Pannu, Christopher Ré, John St. John, Jeremy Sullivan, Joseph Tey, Ben Viggiano, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B. Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Thomas McGrath, Kimberly Powell, Sudarshan Pinglay, Dave P. Burke,

Hani Goodarzi, Patrick D. Hsu, and Brian L. Hie. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652(8112):1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

David H. Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 773–782. PMLR, 2019. URL https://proceedings.mlr.press/v97/brookes19a.html.

Martin Buttenschoen, Garrett M. Morris, and Charlotte M. Deane. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024. doi: 10.1039/D3SC04185A.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5453–5512. PMLR, 2024. URL https://proceedings.mlr.press/v235/campbell24a.html.

Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow matching for controllable biological sequence design, 2025. URL https://arxiv.org/abs/2505.07086.

Yan Chen, Xueru Wang, Xiaobin Deng, Yilun Liu, Xi Chen, Yunwei Zhang, Lei Wang, and Hang Xiao. MatterGPT: A generative transformer for multi-property inverse design of solid-state materials, 2024. URL https://arxiv.org/abs/2408.07608.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2209.14687.

Lucas Ferreira DaSilva, Simon Senan, Judith F. Kribelbauer-Swietek, Zain Munir Patel, Lithin Karmel Louis, Aniketh Janardhan Reddy, Sameer Gabbita, Jonathan D. Rosen, Zach Nussbaum, César Miguel Valdez Córdova, Aaron Wenteler, Noah Weber, Tin M. Tunjic, Martino Mansoldo, Talha Ahmad Khan, Gue-Ho Hwang, Vincent Gardeux, David T. Humphreys, Cameron Smith, Matei Bejan, Peter Bromley, Will Connell, Bart Deplancke, Michael I. Love, Emily S. Wong, Wouter Meuleman, and Luca Pinello. Designing synthetic regulatory elements using the generative AI framework DNA-Diffusion. *Nature Genetics*, 58(1):180–194, 2026. doi: 10.1038/s41588-025-02441-6.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M. Bronstein, Martin Steinegger, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=qmCpJtFZra. Oral.

Qixiu Du, May Nee Poon, Xiaocheng Zeng, Pengcheng Zhang, Zheng Wei, Haochen Wang, Ye Wang, Lei Wei, and Xiaowo Wang. Synthetic promoter design in Escherichia coli based on multinomial diffusion model. *iScience*, 27(11):111207, 2024. doi: 10.1016/j.isci.2024.111207.

Ian Dunn and David R. Koes. FlowMol3: Flow matching for 3D de novo small-molecule generation. *Digital Discovery*, 5(5):2052–2066, 2026. doi: 10.1039/D5DD00363F.

Shikun Feng, Yuyan Ni, Yan Lu, Zhi-Ming Ma, Wei-Ying Ma, and Yanyan Lan. UniGEM: A unified approach to generation and property prediction for molecules. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/223935759d7743c85318639b560882a1-Abstract-Conference.html.

Jenna C Fromer, David E Graff, and Connor W Coley. Pareto optimization to accelerate multi-objective virtual screening. *Digital Discovery*, 3(3):467–481, 2024. doi: 10.1039/D3DD00227F.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, pages 84032–84054, 2025. doi: 10.52202/085713-2534. URL https://papers.neurips.cc/paper_files/paper/2025/hash/6d13e085b79d454da5910e4ca82a3d9d-Abstract-Conference.html.

Erik Hartman, Jonas Wallin, Johan Malmström, and Jimmy Olsson. Controllable protein design with particle-based Feynman-Kac steering, 2025. URL https://arxiv.org/abs/2511.09216.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J. Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c355566ce402de341c3320cf69a10750-Abstract-Conference.html.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. doi: 10.48550/arXiv.2207.12598. URL https://arxiv.org/abs/2207.12598.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas M. Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=tAtpSwjsCB.

Peter Holderrieth, Uriel Singer, Tommi Jaakkola, Ricky T. Q. Chen, Yaron Lipman, and Brian Karrer. GLASS flows: Efficient inference for reward alignment of flow and diffusion models. In *International Conference on Learning Representations (ICLR)*, pages 106232–106266, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ad6ac52e02e14e0f2ddc0d4587ab47f1-Abstract-Conference.html. Oral. Preprint circulated as "GLASS Flows: Transition Sampling for Alignment of Flow and Diffusion Models".

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning (ICML 2022)*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 2022. URL https://proceedings.mlr.press/v162/hoogeboom22a.html.

Qiaoyu Hu, Changzhi Sun, Huan He, Jiazheng Xu, Danlin Liu, Wenqing Zhang, Sumeng Shi, Kai Zhang, and Honglin Li. Target-aware 3D molecular generation based on guided equivariant diffusion. *Nature Communications*, 16(1):7928, 2025. doi: 10.1038/s41467-025-63245-0.

Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. Learning joint 2-D and 3-D graph diffusion models for complete molecule generation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11857–11871, 2024. doi: 10.1109/TNNLS.2024.3416328.

John B. Ingraham, Max Baranov, Zak Costello, Karl W. Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M. Lord, Christopher Ng-Thow-Hing, Erik R. Van Vlack, Shan Tie, Vincent Xue, Sarah C. Cowles, Alan Leung, João V. Rodrigues, Claudio L. Morales-Perez, Alex M. Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V. Panwar, Fritz Obermeyer, Adam R. Root, Andrew L. Beam, Frank J. Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023. doi: 10.1038/s41586-023-06728-8.

Ross Irwin, Alessandro Tibo, Jon Paul Janet, and Simon Olsson. SemlaFlow – efficient 3D molecular generation with latent attention and equivariant flow matching. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pages 3772–3780. PMLR, 2025. URL https://proceedings.mlr.press/v258/irwin25a.html.

Yue Jian, Curtis Wu, Danny Reidenbach, and Aditi S. Krishnapriyan. General binding affinity guidance for diffusion models in structure-based drug design. *Journal of Chemical Information and Modeling*, 66(3):1342–1352, 2026. doi: 10.1021/acs.jcim.5c01166.

Wen-Jie Jiang, Congcong Hu, Futing Lai, Weixiong Pang, Xinyao Yi, Qianyi Xu, Haojie Wang, Jialu Zhou, Hanwen Zhu, Chunge Zhong, Zeyu Kuang, Ruiqi Fan, Jing Shen, Xiaorui Zhou, Yu-Juan Wang, Catherine C. L. Wong, Xiaoqi Zheng, and Hua-Jun Wu. Assessing base-resolution DNA mechanics on the genome scale. *Nucleic Acids Research*, 51(18):9552–9566, 2023. doi: 10.1093/nar/gkad720.

Chaitanya K Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop Sriram, and Zachary W Ulissi. All-atom diffusion transformers: Unified generative modelling of molecules and materials. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 28393–28417. PMLR, 2025. URL https://proceedings.mlr.press/v267/joshi25a.html.

Amit Kadan, Kevin Ryczko, Erika Lloyd, Adrian Roitberg, and Takeshi Yamazaki. Guided multi-objective generative AI to enhance structure-based drug design. *Chemical Science*, 16(29): 13196–13210, 2025. doi: 10.1039/D5SC01778E.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c204d12afa0175285e5aac65188808b4-Abstract-Conference.html.

Haowei Lin, Shanda Li, Haotian Ye, Yiming Yang, Stefano Ermon, Yitao Liang, and Jianzhu Ma. TFG-Flow: Training-free guidance in multimodal generative flow. In *International Conference on Learning Representations*, pages 69043–69078, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/aca042ae71283d25b3d063fc011db9e8-Abstract-Conference.html.

Luca Mariani, Xiao Liu, Kwangwoon Lee, Stephen S Gisselbrecht, Philip A Cole, and Martha L Bulyk. DNA bendability regulates transcription factor binding to nucleosomes. *Nature Structural & Molecular Biology*, 32(11):2185–2195, 2025. doi: 10.1038/s41594-025-01633-2.

Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3D molecule generation and optimization. *Communications Chemistry*, 7(1):150, 2024. doi: 10.1038/s42004-024-01233-z.

Eric Nguyen, Michael Poli, Matthew G. Durrant, Brian Kang, Dhruva Katrekar, David B. Li, Liam J. Bartie, Armin W. Thomas, Samuel H. King, Garyk Brixi, Jeremy Sullivan, Madelena Y. Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A. Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D. Hsu, and Brian L. Hie. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, 2024. doi: 10.1126/science.ado9336.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XsgHl54yO7.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, Yehlin Cho, Kourosh H. Ghamary, Laura Vinué, Brahm J. Yachnin, Andrew M. Wollacott, Stephen Buckley, Adrie H. Westphal, Simon Lindhoud, Sandrine Georgeon, Casper A. Goverde, Georgios N. Hatzopoulos, Pierre Gönczy, Yannick D. Muller, Gerald Schwank, Daan C. Swarts, Alex J. Vecchio, Bernard L. Schneider, Sergey Ovchinnikov, and Bruno E. Correia. One-shot design of functional protein binders with BindCraft. *Nature*, 646(8084):483–492, 2025. doi: 10.1038/s41586-025-09429-6.

Jonghan Park, Galina Prokopchuk, Andrew R. Popchock, Jingzhou Hao, Ting-Wei Liao, Sophia Yan, Dylan J. Hedman, Joshua D. Larson, Brandon K. Walther, Nicole A. Becker, Aakash Basu, L. James Maher, III, Richard J. Wheeler, Charles L. Asbury, Sue Biggins, Julius Lukeš, and Taekjip

Ha. Probing mechanical selection in diverse eukaryotic genomes through accurate prediction of 3D DNA mechanics. *bioRxiv*, 2024. doi: 10.1101/2024.12.22.629997.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael S. Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=K5gV8Yptne.

Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S. Albergo. Discrete flow maps, 2026b. URL https://arxiv.org/abs/2604.09784.

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

Amirmojtaba Sabour, Michael S. Albergo, Carles Domingo-Enrich, Nicholas M. Boffi, Sanja Fidler, Karsten Kreis, and Eric Vanden-Eijnden. Test-time scaling of diffusions with flow maps, 2025. URL https://arxiv.org/abs/2511.22688.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR 2022)*, 2022. URL https://arxiv.org/abs/2202.00512.

Anirban Sarkar, Alejandra Duran, Yiyang Yu, Da-Wei Lin, Yijie Kang, Nirali Somia, Pablo Mantilla, Jessica Zhou, Masayuki Nagai, Ziqi Tang, Kaarina Hanington, Kenneth Chang, and Peter K. Koo. Designing DNA with tunable regulatory activity using discrete diffusion. *bioRxiv*, 2026. doi: 10.1101/2024.05.23.595630. URL https://www.biorxiv.org/content/10.1101/2024.05.23.595630v3. Preprint v3, 27 April 2026.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32483–32498. PMLR, 2023a. URL https://proceedings.mlr.press/v202/song23k.html.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML 2023)*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252. PMLR, 2023b. URL https://proceedings.mlr.press/v202/song23a.html.

Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3D molecule generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 549–568, 2023c. doi: 10.52202/075280-0026. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/01d64478381c33e29ed611f1719f5a37-Abstract-Conference.html.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46495–46513. PMLR, 2024. URL https://proceedings.mlr.press/v235/stark24b.html.

Sophia Tang, Yinuo Zhang, Alexander Tong, and Pranam Chatterjee. Gumbel-Softmax flow matching with straight-through guidance for controllable biological sequence generation, 2025. URL https://arxiv.org/abs/2503.17361.

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and DNA design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 60515–60529. PMLR, 2025. URL https://proceedings.mlr.press/v267/uehara25a.html.

Carlos Vonessen, Charles Harris, Miruna Cretu, and Pietro Liò. TABASCO: A fast, simplified model for molecular generation with improved physical quality. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=Kg6CSrbXl4.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

Minkai Xu, Alexander S. Powers, Ron O. Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3D molecule generation. In *International Conference on Machine Learning (ICML 2023)*, volume 202 of *Proceedings of Machine Learning Research*, pages 38592–38610. PMLR, 2023. URL https://proceedings.mlr.press/v202/xu23n.html.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 22370–22417, 2024. doi: 10.52202/079017-0704. URL https://papers.neurips.cc/paper_files/paper/2024/hash/2818054fc6de6dacdda0f142a3475933-Abstract-Conference.html.

Venkata Rajesh Yella, Devesh Bhimsaria, Debostuti Ghoshdastidar, José A Rodríguez-Martínez, Aseem Z Ansari, and Manju Bansal. Flexibility and structure of flanking DNA impact transcription factor affinity for its core motif. *Nucleic Acids Research*, 46(22):11883–11897, 2018. doi: 10.1093/nar/gky1057.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23117–23127, 2023. doi: 10.1109/ICCV51070.2023.02118. URL https://ieeexplore.ieee.org/document/10377605/.

Cheng Zeng, Jirui Jin, Connor Ambrose, George Karypis, Mark Transtrum, Ellad B. Tadmor, Richard G. Hennig, Adrian Roitberg, Stefano Martiniani, and Mingjie Liu. PropMolFlow: property-guided molecule generation with geometry-complete flow matching. *Nature Computational Science*, 6(3):233–242, 2026. doi: 10.1038/s43588-025-00946-y.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025. doi: 10.1038/s41586-025-08628-5.

Pengcheng Zhang, Haochen Wang, Hanwen Xu, Lei Wei, Liyang Liu, Zhirui Hu, and Xiaowo Wang. Deep flanking sequence engineering for efficient promoter design using DeepSEED. *Nature Communications*, 14(1):6309, 2023. doi: 10.1038/s41467-023-41899-y.

Jia Zheng, Hai-Cheng Yi, and Zhu-Hong You. Equivariant 3D-conditional diffusion model for de novo drug design. *IEEE Journal of Biomedical and Health Informatics*, 29(3):1805–1816, 2025. doi: 10.1109/JBHI.2024.3491318.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 78651–78686. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25c.html.

Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, Christos Skrekas, Vykintas Jauniskis, Nora K. Speicher, Christoph S. Börlin, Vilhelm Verendel, Morteza Haghir Chehreghani, Devdatt Dubhashi, Verena Siewers, Florian David, Jens Nielsen, and Aleksej Zelezniak. Controlling gene expression with deep generative design of regulatory DNA. *Nature Communications*, 13(1):5099, 2022. doi: 10.1038/s41467-022-32818-8.

A Additional Background and Related Work

For small molecule design, unconditional generation is rarely sufficient. Molecules are usually sought under task-specific constraints or desired properties. Materials applications may require compounds with target electronic, thermodynamic, or mechanical properties, such as a desired band gap, formation energy, or elastic response, while drug-discovery applications often require ligands that bind a protein pocket and simultaneously satisfy constraints such as affinity, selectivity, solubility, toxicity, and synthetic accessibility (Zeni et al., 2025; Chen et al., 2024; Kadan et al., 2025; Hu et al., 2025; Fromer et al., 2024). Since these objectives can compete with one another, practical molecular design requires generative models that can adapt across multiple design tasks.

Sequence-dependent DNA mechanics are increasingly recognized as contributors to nucleosome organization and cis-regulatory function (Basu et al., 2021; 2022; Jiang et al., 2023; Mariani et al., 2025). Promoters and active enhancers often coincide with nucleosome-depleted or accessible chromatin, and evidence suggests that these patterns may be shaped in part by sequence-encoded mechanical properties, including intrinsic cyclizability, a proxy for DNA bendability (Basu et al., 2021; Jiang et al., 2023). Yet many motif-centric synthetic regulatory designs specify transcription factor binding sites while leaving the surrounding DNA-shape and flexibility context under-specified (Yella et al., 2018; Zhang et al., 2023). Designing sequences with target mechanical profiles could help synthetic regulatory regions better resemble natural cis-regulatory elements.

Generative models for molecular and sequence design. Recent 3D molecular generators use equivariant diffusion, flow matching, or diffusion-transformer architectures to jointly model atom types and molecular geometry without relying on a fixed autoregressive atom-generation order (Hoogeboom et al., 2022; Zheng et al., 2025; Irwin et al., 2025; Dunn and Koes, 2026; Joshi et al., 2025). Methods for 3D molecule generation, optimization, and property-guided design include PropMolFlow (Zeng et al., 2026), JODO (Huang et al., 2024), GCDM (Morehead and Cheng, 2024), GeoLDM (Xu et al., 2023), EquiFM (Song et al., 2023c), and EEGSDE (Bao et al., 2023).

In parallel, generative models have been developed for regulatory DNA and genome-scale sequence design, including conditional regulatory-sequence diffusion models such as DNA-Diffusion (DaSilva et al., 2026) and D3 (Sarkar et al., 2026), promoter-focused diffusion models such as multinomial diffusion for promoter design (Du et al., 2024), and genome-scale autoregressive foundation models such as Evo (Nguyen et al., 2024) and Evo 2 (Brixi et al., 2026). However, these approaches differ in how controllability is achieved: some models are trained on paired data samples, whereas genome-scale foundation models are often steered by inference-time search. Dirichlet Flow Matching (Stark et al., 2024) develops flow matching on the simplex for categorical sequence generation and enables both classifier and classifier-free guidance. While this provides an important route to controllable DNA generation, our approach differs in that steering is formulated through posterior samples of the clean object under a fixed base generator, allowing arbitrary downstream rewards to be changed at inference-time without retraining the generator or learning a new conditional model.

Inference-time guidance for biological sequences. Existing inference-time methods include classifier or reward guidance, diffusion posterior sampling, flow-map trajectory tilting, Feynman–Kac steering, best-of-N sampling, beam search, and Monte Carlo tree search (Ho and Salimans, 2022; Chung et al., 2023; Lin et al., 2025; Sabour et al., 2025; Hartman et al., 2025; Didi et al., 2026). Recent biological applications include beam-search-guided DNA generation with Evo 2 and inference-time search for protein binder design (Brixi et al., 2026; Didi et al., 2026). These methods typically use particle weights given by rewards estimated by rolling out to the endpoint, denoiser approximations, or task-specific search procedures. In contrast, MFM-based steering estimates the terminal value gradient from posterior samples of the clean object, giving a gradient-informed steering mechanism that keeps the base generator fixed and can be applied across continuous, discrete, and mixed discrete–continuous modalities.

B Additional DNA Evaluation

B.1 Cyclizability, diversity, and sequence-model scores

We evaluated whether C0-steered yeast DNA dMFM samples preserve raw sequence diversity relative to held-out real yeast sequences. The reference data consist of length-50 yeast middle-window DNA sequences over $\{A, C, G, T\}$ from Basu et al. (2021). Samples from the parent-disjoint L = 50 dMFM trained on the 230 a-parents, novelty and nearest-neighbour distances are measured against the 58 held-out test parents. For each steering direction, we analyzed 1,000 unguided and 1,000 guided samples from the clean-split dMFM C0 runs.

For two sequences x, y of length L, normalized Hamming distance was computed as

$$h(x, y) = \frac{1}{L} \sum_{i} \mathbf{1}[x_i \neq y_i].$$

We report exact uniqueness, exact novelty relative to split-train and split-test, nearest-reference distance, nearest-generated distance, and mean pairwise generated distance. Real-data baselines were computed by drawing 1,000 sequences without replacement from the full data, split-train, split-validation, and split-test sets using analyzer seed 0. The primary comparator is the held-out `split_test_sample`. Evo2 was used as an external sequence-model plausibility check. For a sequence $x = (x_1, \ldots, x_L)$, the forward mean log-probability was

$$\text{mean_logprob}_{\text{fwd}}(x) = \frac{1}{L-1} \sum_{i=2}^{L} \log p(x_i \mid x_1, \ldots, x_{i-1}).$$

The same score was computed on the reverse complement $\text{rc}(x)$, and the reported score averages the forward and reverse-complement mean log-probabilities. Negative log-likelihood is reported as $-\text{mean_logprob}$, with nonparametric bootstrap confidence intervals over sequence-level NLL values using 2,000 resamples.

Table 4: Real-data diversity baselines for 1,000-sequence samples.

Set	n	Exact unique	Nearest generated	Pairwise generated	Pairwise median
`train_sample`	1000	1.0000	0.5219	0.7377	0.7400
`split_train_sample`	1000	1.0000	0.5219	0.7373	0.7400
`split_val_sample`	1000	1.0000	0.5184	0.7372	0.7400
`split_test_sample`	1000	1.0000	0.5201	0.7374	0.7400

Table 5: Diversity statistics for generated C0-steered yeast dMFM samples.

Target	Set	n	Guide score mean	Exact unique	Novel train	Novel test	Nearest train	Nearest test	Nearest gen.	Pairwise gen.	Pairwise / held-out	Nearest gen. / held-out
-1.0000	`guided`	1000	-0.9893	1.0000	1.0000	1.0000	0.3966	0.4361	0.4175	0.7010	0.9507	0.8027
-1.0000	`unguided`	1000	-0.1692	1.0000	1.0000	1.0000	0.4449	0.4779	0.5210	0.7364	0.9987	1.0017
1.0000	`guided`	1000	1.0010	1.0000	1.0000	1.0000	0.4446	0.4769	0.4913	0.7372	0.9997	0.9447
1.0000	`unguided`	1000	-0.1766	1.0000	1.0000	1.0000	0.4466	0.4787	0.5215	0.7370	0.9995	1.0027

All generated samples are exact-unique and exact-novel relative to both split-train and split-test. However, exact novelty alone is a weak diversity criterion, so the main comparison is against the held-out `split_test_sample` pairwise and nearest-neighbor baselines. The `target_c0 = +1` guided set preserves held-out-level global diversity: its mean pairwise Hamming fraction is 0.7372, nearly identical to the held-out baseline of 0.7374 and corresponding to $0.9997\times$ the held-out value. Its nearest-generated distance is lower than held-out, 0.4913 versus 0.5201, indicating mild local clustering, but the global pairwise diversity remains effectively held-out-level. By contrast, the `target_c0 = -1` guided set is exact-novel but measurably more clustered, with pairwise diversity 0.7010 or $0.9507\times$ the held-out baseline, and nearest-generated distance 0.4175 or $0.8027\times$ the held-out baseline.

The Evo2 plausibility result is consistent with the diversity result for `target_c0 = +1`: the guided samples have mean NLL 1.3529, close to the held-out `split_test_sample` value of 1.3454 with $\Delta = 0.0075$, and the bootstrap intervals overlap. For `target_c0 = -1`, Evo2 assigns substantially lower NLL than held-out real data, 1.2350 versus 1.3454 with $\Delta = -0.1104$, indicating

Table 6: Evo2 sequence-model plausibility for generated and reference yeast sequences.

Source	Set	Target	n	Bases	Orientation	Evo2 NLL mean [95% CI]	Perplexity	Δ held-out
generated	guided	-1.0000	1000	49	forward_rc_mean	1.2350 [1.2252, 1.2445]	3.4385	-0.1104
generated	unguided	-1.0000	1000	49	forward_rc_mean	1.3613 [1.3584, 1.3641]	3.9013	0.0159
generated	guided	1.0000	1000	49	forward_rc_mean	1.3529 [1.3497, 1.3561]	3.8687	0.0075
generated	unguided	1.0000	1000	49	forward_rc_mean	1.3619 [1.3591, 1.3646]	3.9038	0.0165
reference	split_test_sample	–	1000	49	forward_rc_mean	1.3454 [1.3406, 1.3501]	3.8397	0.0000
reference	split_train_sample	–	1000	49	forward_rc_mean	1.3463 [1.3420, 1.3507]	3.8434	0.0009
reference	split_val_sample	–	1000	49	forward_rc_mean	1.3389 [1.3337, 1.3441]	3.8148	-0.0065
reference	full_sample	–	1000	49	forward_rc_mean	1.3477 [1.3435, 1.3517]	3.8485	0.0023

high sequence-model plausibility. This does not remove the diversity concern, because the same set is measurably clustered under Hamming nearest-neighbor and pairwise-distance statistics. Thus, the supported claim is direction-specific: `target_c0 = +1` guided generation preserves held-out-level pairwise diversity and remains close to held-out real data under Evo2, whereas `target_c0 = −1` remains exact-novel and Evo2-plausible but should not be presented as held-out-diverse.

C Theoretical Background and Proofs

C.1 Flow Matching

The central problem of molecular generative models is to sample the molecule distribution $X_1 \sim p_1(x_1)$. In our molecular experiments, $p_1(x_1)$ is a joint distribution over heavy-atom identities and 3D coordinates. For DNA, $p_1(x_1)$ is a joint categorical distribution over four nucleotide bases $\{A, C, G, T\}$ across the sequence. Each nucleotide is represented by a vertex of $\Delta^3 \subset \mathbb{R}^4$, so a length-$L$ sequence lies in $\mathcal{V}^L$. A perfectly trained base velocity field $b_t(x_t)$ should transport Gaussian noise to this discrete endpoint support in the terminal limit.

Let X_t be a noised state at time t for t between 0 and 1. A common choice is the linear interpolant

$$X_t = (1-t)\, X_0 + t\, X_1, \tag{18}$$

where $X_0 \sim \mathcal{N}(0, \mathbb{I})$ is Gaussian noise at time 0. We denote the distribution of X_t as $p_t(x_t)$. An ODE with the same one-time marginals as the interpolant is

$$dX_t = b_t(X_t)\, dt, \tag{19}$$

where $b_t(X_t)$ is a trained velocity field. The ODE sampler (19) is a special case $\sigma_t = 0$ in the SDE sampler

$$dX_t = (b_t(X_t) + \frac{\sigma_t^2}{2} \nabla \log p_t(X_t))\, dt + \sigma_t\, dB_t. \tag{20}$$

Here σ_t is an arbitrary noise schedule and B_t is the standard Brownian motion. The score $\nabla \log p_t(x_t)$ is related to analytically to the trained velocity field via

$$\nabla \log p_t(x_t) = \frac{t\, b_t(x_t) - x_t}{1-t}. \tag{21}$$

We notice that practical molecular generators have been using the ODE path (19) for efficiency and numerical stability.

C.2 Discrete Meta Flow Maps (dMFM)

Proposition 2 (Diagonal Meta Denoiser). *For linear interpolant* (18)*, the meta denoiser satisfies*

$$\mathbf{\Psi}_{s,s}(\bar{x}_s\,;\, t, x_t) = \psi_{t^*}(S(\bar{x}_s, x_t)), \tag{22}$$

where ψ is the denoiser of the base discrete flow matching model. The base denoiser is evaluated at t^* which is the GLASS reparameterization of t and s. $S(\bar{x}_s, x_t)$ is the GLASS sufficient statistics.

Proof. For the linear interpolant

$$X_t = (1-t)X_0 + tX_1, \tag{23}$$

the base discrete flow matching velocity and denoiser satisfy

$$b_t(x) = \frac{\psi_t(x) - x}{1-t}. \tag{24}$$

In the notation of Holderrieth et al. (2026b), the inner conditional flow path has

$$\bar{\alpha}_s = s, \qquad \bar{\sigma}_s = 1 - s, \qquad \bar{\gamma} = 0, \tag{25}$$

from which we can compute the GLASS weights as:

$$\begin{aligned} w_1(s) &= \frac{\partial_s \bar{\sigma}_s}{\bar{\sigma}_s} = -\frac{1}{1-s}, \\ w_2(s) &= \partial_s \bar{\alpha}_s - \bar{\alpha}_s w_1(s) = 1 + \frac{s}{1-s} = \frac{1}{1-s}, \\ w_3(s) &= -\bar{\gamma} w_1(s) = 0. \end{aligned}$$

By Holderrieth et al. (2026b), the GLASS denoiser is related to the denoiser of the base flow with sufficient statistics $S(\bar{x}_s, x_t)$ as

$$D_{\mu(s),\Sigma(s)}(x_t, \bar{x}_s) = \psi_{t^*}(S(\bar{x}_s, x_t)), \tag{26}$$

as the GLASS velocity is therefore

$$\begin{aligned} \bar{b}_s(\bar{x}_s; t, x_t) &= w_1(s)\bar{x}_s + w_2(s)\psi_{t^*}(S(\bar{x}_s, x_t)) + w_3(s)x_t \\ &= -\frac{1}{1-s}\bar{x}_s + \frac{1}{1-s}\psi_{t^*}(S(\bar{x}_s, x_t)) \\ &= \frac{\psi_{t^*}(S(\bar{x}_s, x_t)) - \bar{x}_s}{1-s}. \end{aligned} \tag{27}$$

The base discrete flow denoiser evaluated at time $t^*$is related to the GLASS velocity as

$$\psi_{t^*}(S(\bar{x}_s, x_t)) = (1-s)\bar{b}_s(\bar{x}_s; t, x_t) + \bar{x}_s. \tag{28}$$

Now we identify $\bar{b}_s$ with the diagonal meta velocity $\mathbf{v}_{s,s}$. By the definition of the discrete meta flow map in equation (8), $\mathbf{X}_{s,u}(\bar{x}_s; t, x_t)$ is the solution operator of the inner ODE equation (7). Hence, for fixed (t, x_t) and initial condition $\bar{x}_s$,

$$\frac{d}{du}\mathbf{X}_{s,u}(\bar{x}_s; t, x_t)\Big|_{u=s} = \bar{b}_s(\bar{x}_s; t, x_t). \tag{29}$$

On the other hand, the discrete meta flow map is parametrized with the meta-denoiser as,

$$\mathbf{X}_{s,u}(\bar{x}_s; t, x_t) = \frac{1-u}{1-s}\bar{x}_s + \frac{u-s}{1-s}\mathbf{\Psi}_{s,u}(\bar{x}_s; t, x_t). \tag{30}$$

Differentiating equation (30) with respect to u gives

$$\frac{d}{du}\mathbf{X}_{s,u}(\bar{x}_s; t, x_t) = -\frac{1}{1-s}\bar{x}_s + \frac{1}{1-s}\mathbf{\Psi}_{s,u}(\bar{x}_s; t, x_t) + \frac{u-s}{1-s}\partial_u \mathbf{\Psi}_{s,u}(\bar{x}_s; t, x_t). \tag{31}$$

On the diagonal of meta flow map times $u = s$, the last term vanishes, and we have

$$\frac{d}{du}\mathbf{X}_{s,u}(\bar{x}_s; t, x_t)\Big|_{u=s} = \frac{\mathbf{\Psi}_{s,s}(\bar{x}_s; t, x_t) - \bar{x}_s}{1-s}. \tag{32}$$

By the definition of the meta velocity and equation (29), we have

$$\mathbf{v}_{s,s}(\bar{x}_s; t, x_t) = \frac{\mathbf{\Psi}_{s,s}(\bar{x}_s; t, x_t) - \bar{x}_s}{1-s} = \bar{b}_s(\bar{x}_s; t, x_t). \tag{33}$$

Combining equation (33) with (28), we obtain

$$\mathbf{\Psi}_{s,s}(\bar{x}_s; t, x_t) = (1-s)\mathbf{v}_{s,s}(\bar{x}_s; t, x_t) + \bar{x}_s = (1-s)\bar{b}_s(\bar{x}_s; t, x_t) + \bar{x}_s = \psi_{t^*}(S(\bar{x}_s, x_t)). \tag{34}$$

□

For the independent Gaussian noises in the outer and inner linear interpolants, the promised reparameterization is explicit. For $0 \le t, s < 1$, let

$$q = \frac{t^2}{(1-t)^2} + \frac{s^2}{(1-s)^2}, \qquad t^* = \frac{\sqrt{q}}{1+\sqrt{q}}, \qquad S(\bar{x}_s, x_t) = \frac{\frac{t}{(1-t)^2}x_t + \frac{s}{(1-s)^2}\bar{x}_s}{\sqrt{q}(1+\sqrt{q})}. \tag{35}$$

Indeed, the two Gaussian likelihoods depend on the endpoint through precision q and the numerator in S. The single observation at t^* has the same likelihood in that endpoint, up to an endpoint-independent factor, so its posterior mean is the required conditional denoiser. At $s = t = 0$, the conditional denoiser is the unconditional mean, so no value of S is needed. The argument also applies to endpoints supported on simplex vertices and to jointly modeled modalities with a common schedule.

dMFM is the solution operator of an ODE in equation (8) imposes further constraints on $\mathbf{X}_{s,u}$ and the meta denoiser $\mathbf{\Psi}_{s,u}(\bar{x}_s\,;\,t, x_t)$. One such constraint is $\mathbf{X}_{s,u}$ obeying the Euler's Equation:

$$\partial_s \mathbf{X}_{s,u}(\bar{x}_s; t, x_t) + \mathbf{v}_{s,s}(\bar{x}_s; t, x_t) \cdot \nabla \mathbf{X}_{s,u}(\bar{x}_s; t, x_t) = 0. \tag{36}$$

Expressing 36 in terms of the meta denoiser $\mathbf{\Psi}$ similar to Potaptchik et al. (2026b), we have:

$$\mathbf{\Psi}_{s,w}(\bar{x}_s; t, x_t) = \mathbf{\Psi}_{s,s}(\bar{x}_s; t, x_t) - \gamma_{w,s}\big(\partial_s \mathbf{\Psi}_{s,w}(\bar{x}_s; t, x_t) + J_x \mathbf{\Psi}_{s,w}(\bar{x}_s; t, x_t)\bar{b}_s(\bar{x}_s; t, x_t)\big). \tag{37}$$

Here $\gamma_{w,s} = \frac{(s-w)(1-s)}{1-w}$ and J_x is the Jacobian acting on $\mathbf{\Psi}$. The right side of constraint (37) is an identity for the exact denoiser, but need not be a probability vector for a neural approximation. A simplex-valued *Eulerian Self Distillation Target* $\mathbf{T}^{\mathrm{ESD}}_{s,w}$ is therefore required in the KL loss below; a logit-space construction is given below.

$$\mathcal{L}_{\mathrm{ESD}}(\hat{\mathbf{\Psi}}) = \mathbb{E}\left[D_{KL}\left(\mathrm{sg}(\mathbf{T}^{\mathrm{ESD}}_{s,w}(\hat{\mathbf{\Psi}})(\bar{x}_s\,;\,t, x_t))\middle\|\hat{\mathbf{\Psi}}_{s,w}(\bar{x}_s\,;\,t, x_t)\right)\right]. \tag{38}$$

Constraint (37) has two equivalent exact-flow formulations, discussed below. These identities do not imply identical optimization behavior for finite neural networks. The Progressive Self Distillation Loss $\mathcal{L}_{\mathrm{PSD}}$ and Lagrangian Self Distillation Loss $\mathcal{L}_{\mathrm{LSD}}$ are defined next. One such constraint comes from the semi-group property of $\mathbf{X}_{s,u}$. We require applying one dMFM from time s to w is equivalent to iterative applications of dMFM from time s to u and then from u to w for $s \le u \le w$, i.e.

$$\mathbf{X}_{s,w}(\bar{x}_s\,;\,t, x_t) = \mathbf{X}_{u,w}(\mathbf{X}_{s,u}(\bar{x}_s\,;\,t, x_t);\,t, x_t) \tag{39}$$

Converting equation (39) from meta flow map $\mathbf{X}_{s,u}$ to denoiser $\mathbf{\Psi}_{s,u}$ we get:

$$\mathbf{\Psi}_{s,w}(\bar{x}_s\,;\,t, x_t) = \alpha_{s,u,w}\mathbf{\Psi}_{s,u}(\bar{x}_s\,;\,t, x_t) + \beta_{s,u,w}\mathbf{\Psi}_{u,w}(\mathbf{X}_{s,u}(\bar{x}_s\,;\,t, x_t)\,;\,t, x_t) \tag{40}$$

Here

$$\alpha_{s,u,w} = \frac{(u-s)(1-w)}{(w-s)(1-u)}, \qquad \beta_{s,u,w} = \frac{(w-u)(1-s)}{(w-s)(1-u)},$$

for $s < u < w < 1$, with boundary values given by limits. These weights are nonnegative and sum to one. We call the right side of equation (40) the *Progressive Self Distillation Target* $\mathbf{T}^{\mathrm{PSD}}_{s,u,w}(\mathbf{\Psi})(\bar{x}_s\,;\,t, x_t)$. The neural network $\hat{\mathbf{\Psi}}_{s,w}$ is trained to match the target $\mathbf{T}^{\mathrm{PSD}}_{s,u,w}(\hat{\mathbf{\Psi}})$ for any $s \le u \le w$ according to the loss (40)

$$\mathcal{L}_{\mathrm{PSD}}(\hat{\mathbf{\Psi}}) = \mathbb{E}\left[D_{KL}\left(\mathrm{sg}(\mathbf{T}^{\mathrm{PSD}}_{s,u,w}(\hat{\mathbf{\Psi}})(\bar{x}_s\,;\,t, x_t))\middle\|\hat{\mathbf{\Psi}}_{s,w}(\bar{x}_s\,;\,t, x_t)\right)\right] \tag{41}$$

The above loss is defined as the *Progressive Self Distillation Loss (PSD)*. There is another constraint, equivalent to constraint (40). Similar to Potaptchik et al. (2026b), it is:

$$\mathbf{\Psi}_{s,w}(\bar{x}_s; t, x_t) = \mathbf{\Psi}_{w,w}(\mathbf{X}_{s,w}(\bar{x}_s; t, x_t); t, x_t) - \gamma_{s,w}\,\partial_w \mathbf{\Psi}_{s,w}(\bar{x}_s; t, x_t), \tag{42}$$

Here $\gamma_{s,w} = (w-s)(1-w)/(1-s)$. As for ESD, the raw right side need not remain on the simplex for a neural approximation. The KL loss therefore requires a simplex-valued *Lagrangian Self Distillation Target* $\mathbf{T}^{\mathrm{LSD}}_{s,w}(\mathbf{\Psi})(\bar{x}_s; t, x_t)$. At training time, constraint (42) is enforced via the KL-divergence

$$\mathcal{L}_{\mathrm{LSD}}(\hat{\mathbf{\Psi}}) = \mathbb{E}\left[D_{KL}\left(\mathrm{sg}(\mathbf{T}^{\mathrm{LSD}}_{s,w}(\hat{\mathbf{\Psi}})(\bar{x}_s\,;\,t, x_t))\middle\|\hat{\mathbf{\Psi}}_{s,w}(\bar{x}_s\,;\,t, x_t)\right)\right], \tag{43}$$

which is defined as the *Lagrangian Self Distillation Loss (LSD)*.

Simplex-valued consistency targets. For ESD, write $D_s = \partial_s + \bar{b}_s \cdot \nabla_x$, $\kappa_{s,w} = (1-w)/((w-s)(1-s))$, and

$$\delta_{s,w} = D_s \hat{\mathbf{z}}_{s,w} - \langle \hat{\mathbf{\Psi}}_{s,w}, D_s \hat{\mathbf{z}}_{s,w} \rangle \mathbf{1}.$$

When the logarithm has positive arguments, the corresponding logit-space target is

$$\mathbf{T}^{\mathrm{ESD}}_{s,w} = \mathrm{softmax}\big(\hat{\mathbf{z}}_{s,s} - \log(\mathbf{1} - \kappa^{-1}_{s,w}\delta_{s,w})\big)\,.$$

For LSD, put $\eta_{s,w} = \partial_w \hat{\mathbf{z}}_{s,w} - \langle \hat{\mathbf{\Psi}}_{s,w}, \partial_w \hat{\mathbf{z}}_{s,w} \rangle \mathbf{1}$. The analogous target is

$$\mathbf{T}^{\mathrm{LSD}}_{s,w} = \mathrm{softmax}\Big(\hat{\mathbf{z}}_{w,w}(\hat{\mathbf{X}}_{s,w}) - \log(\mathbf{1} + \gamma_{s,w}\eta_{s,w})\Big)\,.$$

The fixed outer condition (t, x_t) is suppressed in these expressions. These are formal logit-space targets. In practice, we use clipping and weighting of the separate loss terms in the training.

C.3 Composite Rewards

For composite rewards, the relevant independence is under the conditional endpoint law $p_{1|t}(\cdot \mid x_t)$. Conditional independence yields additivity of value functions, and hence of their gradients where the identity holds in a neighborhood. Gradient additivity alone does not imply independence.

Proposition 3 (Composite Reward). *Fix t and suppose the conditional joint moment-generating function of $(r_1(X_1), r_2(X_1))$ is finite in a neighborhood of the origin. At each x_t, these rewards are independent under $p_{1|t}(\cdot \mid x_t)$ if and only if*

$$V_t^{w_1 r_1 + w_2 r_2}(x_t) = V_t^{w_1 r_1}(x_t) + V_t^{w_2 r_2}(x_t)$$

for all weights in that neighborhood. If this identity holds on an open set of conditioning states and the value functions are differentiable there, their gradients add on that set.

Remark: Distinct predictors or physical properties do not establish conditional independence. Without it, apply the estimator to the combined reward. The finite-sample estimator (3) is generally biased; the estimator statement in the proof below is conditional on genuinely unbiased individual estimators being available.

Proof. Let t and x_t be fixed. Throughout the proof, all expectations are taken with respect to the conditional law $X_1 \sim p_{1|t}(\cdot \mid x_t)$. Define

$$A := r_1(X_1), \qquad B := r_2(X_1).$$

For any reward ρ, we denote

$$V_t^{\rho}(x_t) := \log \mathbb{E}\left[\exp(\rho(X_1)) \mid X_t = x_t\right],$$

and let the estimator $\widehat{\nabla V}_t^{\,\rho}(x_t)$ estimates $\nabla V_t^{\rho}(x_t)$. Suppose that $A = r_1(X_1)$ and $B = r_2(X_1)$ are independent random variables under $p_{1|t}(\cdot \mid x_t)$. By the moment-generating-function factorization criterion for independence, or equivalently by the uniqueness theorem for moment generating functions, independence of A and B implies that for all a, b in a neighborhood of the origin,

$$\mathbb{E}\left[e^{aA+bB}\right] = \mathbb{E}\left[e^{aA}\right]\,\mathbb{E}\left[e^{bB}\right].$$

For weights $a = w_1$ and $b = w_2$ in this neighborhood, we obtain

$$\begin{aligned} V_t^{w_1 r_1 + w_2 r_2}(x_t) &= \log \mathbb{E}\left[\exp\left(w_1 r_1(X_1) + w_2 r_2(X_1)\right)\right] \\ &= \log\left(\mathbb{E}\left[e^{w_1 r_1(X_1)}\right]\,\mathbb{E}\left[e^{w_2 r_2(X_1)}\right]\right) \\ &= \log \mathbb{E}\left[e^{w_1 r_1(X_1)}\right] + \log \mathbb{E}\left[e^{w_2 r_2(X_1)}\right] \\ &= V_t^{w_1 r_1}(x_t) + V_t^{w_2 r_2}(x_t). \end{aligned} \tag{44}$$

If this identity holds on an open set of x_t and the value functions are differentiable, it gives:

$$\nabla V_t^{w_1 r_1 + w_2 r_2}(x_t) = \nabla V_t^{w_1 r_1}(x_t) + \nabla V_t^{w_2 r_2}(x_t).$$

If the individual estimators are unbiased, linearity of expectation gives

$$\begin{aligned}\mathbb{E}_{\mathrm{MC}}\left[\widehat{\nabla V}_t^{\,w_1 r_1}(x_t) + \widehat{\nabla V}_t^{\,w_2 r_2}(x_t)\right] &= \nabla V_t^{\,w_1 r_1}(x_t) + \nabla V_t^{\,w_2 r_2}(x_t) \\ &= \nabla V_t^{\,w_1 r_1 + w_2 r_2}(x_t).\end{aligned}$$

Hence the sum $\widehat{\nabla V}_t^{\,w_1 r_1}(x_t) + \widehat{\nabla V}_t^{\,w_2 r_2}(x_t)$ is an unbiased estimator of $\nabla V_t^{\,w_1 r_1 + w_2 r_2}(x_t)$.

Conversely, suppose that for all weights w_1, w_2 in a neighborhood of the origin, the composite value function factorizes as

$$V_t^{\,w_1 r_1 + w_2 r_2}(x_t) = V_t^{\,w_1 r_1}(x_t) + V_t^{\,w_2 r_2}(x_t).$$

Exponentiating both sides, we have

$$\mathbb{E}\left[\exp\left(w_1 r_1(X_1) + w_2 r_2(X_1)\right)\right] = \mathbb{E}\left[e^{w_1 r_1(X_1)}\right]\ \mathbb{E}\left[e^{w_2 r_2(X_1)}\right].$$

In term of moment generating functions, the above equations says

$$M_{A,B}(w_1, w_2) = M_A(w_1) M_B(w_2), \tag{45}$$

where $M_{A,B}$ is the joint moment generating function of (A, B) and M_A, M_B are the marginal moment generating functions. By the moment-generating-function factorization criterion for independence, equation (45) implies that A and B are independent. Equivalently, $r_1(X_1)$ and $r_2(X_1)$ are independent random variables under $p_{1|t}(\cdot \mid x_t)$. □

D ALGORITHMS

Algorithm 1 dMFM Training

Require: Initial parameters θ of $\hat{\mathbf{z}}$; base discrete flow matching denoiser ψ

while not converged **do**
  Sample $x_0 \sim p_0$, $x_1 \sim p_1$, $\epsilon \sim \mathcal{N}(0, \mathbb{I}_d)$; sample $t, s, u \sim \text{Unif}[0, 1]$
  Arrange $s < u$
  $x_t = (1-t)x_0 + tx_1$; $\bar{x}_s = (1-s)\epsilon + s\, x_1$
  # Monte Carlo estimate of losses
  Compute GLASS time $t^*(t, s)$ and sufficient statistic $S(\bar{x}_s, x_t)$
  Compute GLASS denoiser $\psi_{t^*}(S(\bar{x}_s, x_t))$
  $\mathcal{L}_{\text{diag}} \leftarrow -\psi_{t^*}(S(\bar{x}_s, x_t)) \log \Big(\text{softmax}(\hat{\mathbf{z}}_{s,s}(\bar{x}_s\,;\, t, x_t)) \Big)$
  Compute ESD Loss $\mathcal{L}_{\text{ESD}}$
  $\mathcal{L}_{\text{dMFM}} \leftarrow \mathcal{L}_{\text{diag}} + \mathcal{L}_{\text{ESD}}$
  Compute $\nabla_\theta \mathcal{L}_{\text{dMFM}}$
  minimize $\mathcal{L}_{\text{dMFM}}$ by gradient descent
end while
return θ

Algorithm 2 Multimodal MFM (multiMFM) Steering

Require: Continuous MFM Φ^c; discrete dMFM $\mathbf{X}^d$; multimodal MFM $\mathbf{X} = (\Phi^c, \mathbf{X}^d)$; base velocity field b_t^y and b_t^a for continuous modality y and discrete modality a; differentiable reward $r(y, a)$; times $0 = t_0 < t_1 < \cdots < t_T = 1$; dimension d_c of y; token size M; number of MFM lookaheads N

Initialize $y_0, a_0 \sim p_0$
for $k = 0, ..., T-1$ **do**
  $dt \leftarrow t_{k+1} - t_k$
  Sample iid $\epsilon_i^y \sim \mathcal{N}(0, \mathbb{I}_{d_c})$ and $\epsilon_i^a \sim \mathcal{N}(0, \mathbb{I}_M)$ for $i = 1, ..., N$
  $(y_{1,i},\, a_{1,i}) \leftarrow \mathbf{X}_{0,1}(\epsilon_i^y,\, \epsilon_i^a\,;\, t_k, y_{t_k}, a_{t_k})$
  Compute $\widehat{\partial_y V}^r_{t_k}(y_{t_k}, a_{t_k})$ and $\widehat{\partial_a V}^r_{t_k}(y_{t_k}, a_{t_k})$ with equation (3)
  Advance to $(y_{t_{k+1}}, a_{t_{k+1}})$ with ODE (16)
end for
return (y_1, a_1)

The property regressor $\mathcal{R}$ in these algorithms is the guide; the held-out oracle is used only for final evaluation.

For multiMFM-SS with the steer-search trick, there is a storage module which stores the look-ahead that has the lowest error $||\mathcal{R}(y_1, a_1) - \mu||$ and is PB-valid. The storage module is rewritten everytime a PB-valid molecule look-ahead with smaller error is found.

Algorithm 3 multiMFM with steer-search for molecule design

Require: Continuous MFM Φ^c; discrete dMFM $\mathbf{X}^d$; multimodal MFM $\mathbf{X} = (\Phi^c, \mathbf{X}^d)$; base velocity field b_t^y and b_t^a for continuous modality y and discrete modality a; differentiable reward $r(y, a)$; times $0 = t_0 < t_1 < \cdots < t_T = 1$; dimension d_c of y; token size M; number of MFM lookaheads N; target property value μ; property regressor $\mathcal{R}$; PB-validity Checker $\pi(y, a) \in \{0, 1\}$ with $\pi = 1$ being PB-valid; scaling factor β.

Set $r(y, a) = -\beta \|\mathcal{R}(y, a) - \mu\|^2$
Initialize $(y_0, a_0) \sim p_0$; search module $S \leftarrow \emptyset$; $\text{err}(S) \leftarrow +\infty$
for $k = 0, ..., T - 1$ **do**
  $dt \leftarrow t_{k+1} - t_k$
  Evaluate the chosen guidance schedule with its endpoint regularization
  Sample iid $\epsilon_i^y \sim \mathcal{N}(0, \mathbb{I}_{d_c})$ and $\epsilon_i^a \sim \mathcal{N}(0, \mathbb{I}_M)$ for $i = 1, ..., N$
  $(y_{1,i},\ a_{1,i}) \leftarrow \mathbf{X}_{0,1}(\epsilon_i^y,\ \epsilon_i^a\ ;\ t_k, y_{t_k}, a_{t_k})$
  for $i = 1, ..., N$ **do**
    Compute $\text{err}_i \leftarrow ||\mathcal{R}(y_{1,i}, a_{1,i}) - \mu||$ and $\pi(y_{1,i}, a_{1,i})$
    if $\text{err}_i < \text{err}(S)$ **and** $\pi(y_{1,i}, a_{1,i}) = 1$ **then**
      $S \leftarrow \{(y_{1,i}, a_{1,i}),\ \text{err}_i\}$
    end if
  end for
  Compute tilt per look-ahead:
    $\text{tilt}_i \leftarrow \exp\big(-\beta \|\mathcal{R}(y_{1,i}, a_{1,i}) - \mu\|^2\big)$
  Compute $\widehat{\partial_y V}^r_{t_k}(y_{t_k}, a_{t_k})$ and $\widehat{\partial_a V}^r_{t_k}(y_{t_k}, a_{t_k})$ with equation (3)
  Advance to $(y_{t_{k+1}}, a_{t_{k+1}})$ with ODE (16)
end for
if $S = \emptyset$ **then**
  return (y_1, a_1)
else if $\text{err}(y_1, a_1) < \text{err}(S)$ **and** $\pi(y_1, a_1)$ is PB-valid **then**
  return (y_1, a_1)
else
  return the molecule stored in S
end if

E ARCHITECTURES

E.1 DMFM

The student model is a 1D DiT-based conditional denoiser that operates on sequence inputs $x \in \mathbb{R}^{B\times L\times K}$ and produces a conditional mean estimate

$$\Psi_{s,t}(x; t_{\text{cond}}, x_{\text{cond}}).$$

Rather than directly predicting velocity, the model induces a flow via

$$v_{s,t}(x) = \frac{\Psi_{s,t}(x) - x}{1 - s}.$$

The architecture follows a standard DiT design with $N = 4$ transformer blocks. Each block consists of AdaLN-modulated self-attention and MLP layers with residual gating. Rotary positional embeddings are applied along the sequence dimension.

The model conditions on three scalars (s, t, t_{cond}) via learned timestep embeddings. Two conditioning streams are used: the first half of the network uses

$$c_{\text{first}} = \text{SiLU}(e_s(s) + e_t(t) + e_{\text{cond}}(t_{\text{cond}})),$$

while the second half and output layer use

$$c_{\text{second}} = \text{SiLU}(e_s^{(2)}(s) + e_t^{(2)}(t) + e_{\text{cond}}(t_{\text{cond}})).$$

Inputs lie in the ambient space over a K-token alphabet and are embedded via a linear combination of learned embeddings $E_x \in \mathbb{R}^{K\times d}$. A separate embedding $E_{x_{\mathrm{cond}}}$ is used for the conditional observation.

The conditional input is gated by t_{cond} through an AdaLN-style modulation:

$$\widetilde{x}_{\mathrm{cond}} = (1+\Gamma_{\mathrm{cond}}) \odot E_{x_{\mathrm{cond}}}(x_{\mathrm{cond}}) + \Delta_{\mathrm{cond}},$$

with $\Delta_{\mathrm{cond}}, \Gamma_{\mathrm{cond}}$ produced from $e_{\mathrm{cond}}(t_{\mathrm{cond}})$. The final token input is

$$x_{\mathrm{tok}} = E_x(x) + \widetilde{x}_{\mathrm{cond}}.$$

For the conditional branch to vanish exactly at $t_{\mathrm{cond}} = 0$, the gate must explicitly enforce that boundary condition; the displayed affine modulation alone does not guarantee it.

Each transformer block applies AdaLN-modulated attention and MLP updates:

$$h \leftarrow h + g_{\mathrm{msa}} \odot \mathrm{MSA}(\cdot), \qquad h \leftarrow h + g_{\mathrm{mlp}} \odot \mathrm{MLP}(\cdot),$$

where modulation parameters are affine functions of the conditioning vector. We trained the dMFM for 10 hours on one 80GB NVIDIA H100 GPU.

Table 7: Hyperparameters for DNA dMFM.

Hyperparameter	**Value**
Data / state	
Sequence length L	50
Alphabet size K	4
Gap range	$[0.75, 1.0]$
Architecture	
Hidden dimension d	192
Transformer blocks	4
Attention heads	6
Head dimension	32
MLP expansion ratio	4.0
Conditioning dimension	192
Attention softcap	50.0
Parameters	
Total parameters	3,420,484
DiT backbone	2,912,836
Embeddings	1,536
x_{cond} gate	74,112
Time embedders	432,000
Training	
GLASS NFE	8
Loss / curriculum	
Consistency loss $\mathcal{L}_{\mathrm{ESD}}$	ESD loss
Diagonal weight	0.35
Consistency weight	10.0

Table 8: Training setup. Specific parameters for the different lengths are specified in Table 9.

E.2 Multimodal Meta Flow Map

E.2.1 Unconditional Flow Matching Model

We model the joint state

$$x_1 = (y_1, a_1) \in \mathbb{R}^d \times \{1, \dots, K\},$$

where a_1 is a discrete class label and y_1 is a continuous coordinate. We embed the discrete variable in one-hot space and train entirely in the Euclidean state

$$x_1 = \big(y_1, \mathrm{OneHot}(a_1)\big) \in \mathbb{R}^d \times \mathbb{R}^K.$$

Table 9: DNA training per sequence length. All models are parent-disjoint (generator and dMFM on the 230 a-training parents, selected on the 29 a-validation parents). Architecture is identical across lengths (d=192, 4 DiT blocks, 6 heads, FF×4); only batch size varies.

	L=50	L=100	L=200	L=400
Base generator (DFM/DiT)				
Training steps	200,000	200,000	200,000	200,000
Batch size	512	256	128	64
Learning rate		5×10^{-5}		
dMFM student				
Selected step	49,000	39,000	45,000	45,750
Batch size	64	64	32	16
Learning rate		1×10^{-4}		
Windows per parent	143	135	121	93

Table 10: Parent-disjoint split. Parents are split *before* windows are extracted, so every window count is an exact whole-parent multiple. The five subsets are mutually parent-disjoint, verified at build time and again at training time. No subsampling is applied to any training consumer.

Subset	Parents	L=50	L=100	L=200	L=400
`a_train` (generator, dMFM, guide)	230	32,890	31,050	27,830	21,390
`a_val` (selection)	29	4,147	3,915	3,509	2,697
`b_train` (oracle)	230	32,890	31,050	27,830	21,390
`b_val` (oracle selection)	29	4,147	3,915	3,509	2,697
`test` (held out)	58	8,294	7,830	7,018	5,394
Total	576	82,368	77,760	69,696	53,568

Classes are recovered only at the end by projecting the discrete component back to a categorical variable. We use a Gaussian-source linear interpolant for both modalities. Given

$$y_0 \sim \mathcal{N}(0, I_d), \qquad a_0 \sim \mathcal{N}(0, I_K),$$

independently of X_1, define

$$y_t = (1-t)y_0 + ty_1, \qquad a_t = (1-t)a_0 + t\,\mathrm{OneHot}(a_1).$$

Thus $a_t \in \mathbb{R}^K$ during training; it is not constrained to the simplex except at the endpoint. The joint dynamics are therefore a smooth ODE in $\mathbb{R}^{d+K}$. The conditional path velocities are

$$v_t^y = \frac{y_1 - y_t}{1-t}, \qquad v_t^a = \frac{\mathrm{OneHot}(a_1) - a_t}{1-t}.$$

Rather than regress velocities directly, we predict the data endpoint. A single network

$$f_\theta(t, y_t, a_t) = \big(\hat{y}_1(t, y_t, a_t), \hat{z}(t, y_t, a_t)\big)$$

outputs a continuous endpoint $\hat{y}_1 \in \mathbb{R}^d$ and class logits $\hat{z} \in \mathbb{R}^K$. The predicted discrete endpoint is

$$\hat{a}_1(t, y_t, a_t) = \mathrm{softmax}\hat{z}(t, y_t, a_t) \in \Delta^{K-1}.$$

The induced sampling velocity is

$$v_\theta(t, y_t, a_t) = \left(\frac{\hat{y}_1 - y_t}{1-t}, \frac{\hat{a}_1 - a_t}{1-t}\right).$$

This parametrization matches the linear interpolant structure, gives both modalities the same endpoint-to-current-state velocity form, and allows the discrete head to be trained with standard cross-entropy. We train with endpoint losses

$$\mathcal{L}_y(\theta) = \mathbb{E}\left[\|\hat{y}_1(t, y_t, a_t) - y_1\|_2^2\right],$$

Table 11: C0 regressor accuracy on the 58 held-out test parents (8,294 windows), parent-disjoint from both regressors' training and selection parents. The guide supplies the steering gradient; the oracle is used only for post-hoc scoring and never enters generation, selection or hyperparameter choice.

Model	Trained on	MAE	RMSE	Pearson r	Spearman ρ
Guide	`a_train` (230 parents)	0.1588	0.2306	0.8774	0.8838
Oracle	`b_train` (230 parents)	0.1620	0.2347	0.8724	0.8764

$$\mathcal{L}_a(\theta) = \mathbb{E}\left[\mathrm{CE}\big(\hat{z}(t, y_t, a_t), a_1\big)\right],$$

and

$$\mathcal{L}(\theta) = \mathcal{L}_y(\theta) + \mathcal{L}_a(\theta).$$

The expectation is over

$$X_1 \sim p_{\mathrm{data}}, \qquad t \sim \mathcal{U}[0,1], \qquad (y_0, a_0) \sim \mathcal{N}(0, I_d) \otimes \mathcal{N}(0, I_K).$$

For the linear interpolant, the MSE endpoint loss is equivalent to velocity matching up to the factor $(1-t)^{-2}$. The cross-entropy loss is a discrete denoising objective whose softmax output is also used as the endpoint prediction in the velocity field. The endpoint predictor is a small DiT-style transformer, similar to that of Potaptchik et al. (2026b), over two modality tokens. We encode

$$h^y = W_y y_t + e_y, \qquad h^a = W_a a_t + e_a,$$

where e_y, e_a are learned modality embeddings, giving

$$H_0 = (h^y, h^a) \in \mathbb{R}^{2\times D}.$$

A stack of transformer blocks with full self-attention lets the continuous and discrete tokens exchange information at every layer. Time enters through adaLN-Zero. A sinusoidal embedding of t is passed through an MLP to produce $\tau(t)$, which generates LayerNorm modulation parameters (s, b, g) for each block. For the MLP sublayer,

$$\mathrm{Block}(H, \tau) = H + g_2(\tau) \odot \mathrm{MLP}\left(\mathrm{LN}_2(H) \odot (1 + s_2(\tau)) + b_2(\tau)\right),$$

with an analogous modulated attention sublayer. The gates $g(\tau)$ are zero-initialized, so the network starts as the identity and learns time-dependent corrections. After the final block, modality-specific heads decode the two tokens:

$$\hat{y}_1 = W_y^{\mathrm{out}} H_L[y], \qquad \hat{z} = W_a^{\mathrm{out}} H_L[a].$$

Thus

$$f_\theta(t, y_t, a_t) = (\hat{y}_1, \hat{z}).$$

At inference, we sample

$$(y_0, a_0) \sim \mathcal{N}(0, I_d) \otimes \mathcal{N}(0, I_K)$$

and integrate

$$\frac{d}{dt}\begin{pmatrix} y_t \\ a_t \end{pmatrix} = v_\theta(t, y_t, a_t)$$

on $[0, 1-\epsilon]$, for example with explicit Euler. The endpoint-form velocities are

$$\frac{\hat{y}_1 - y_t}{1-t}, \qquad \frac{\hat{a}_1 - a_t}{1-t}.$$

At the terminal step, the continuous output is read from $\hat{y}_1$, while the class is sampled from

$$\mathrm{Categorical}(\mathrm{softmax}\hat{z}).$$

The margin $\epsilon > 0$ avoids the singularity at $t = 1$. Hyperparameters found in Table 12.

Table 12: Training hyperparameters for the multiMFM toy example.

Hyperparameter	**Value**
Data / state	
Number of classes K	3
Continuous dimension d	2
Source distribution	$\mathcal{N}(0, I)$ for both modalities
Probability path	Linear interpolant
Architecture	
Model width D	128
Transformer blocks	6
Attention heads	8
MLP expansion ratio	4
Tokens per sample	2, one per modality
Conditioning	adaLN-Zero
Optimization	
Optimizer	AdamW
Peak learning rate	3×10^{-4}
Learning-rate schedule	Cosine decay with warmup
Warmup steps	500
EMA decay	0.999
Training	
Batch size	4096
Training steps	20,000
Loss / curriculum	
Continuous loss $\mathcal{L}_y$	MSE on $\hat{y}_1$
Discrete loss $\mathcal{L}_c$	Cross-entropy on $\hat{z}$
Consistency loss $\mathcal{L}_{\mathrm{ESD}}$	ESD loss
Total loss	$\mathcal{L} = \mathcal{L}_y + \mathcal{L}_c + \mathcal{L}_{\mathrm{ESD}}$
Sampling	
Integrator	Explicit Euler
Sampling steps	100
Early-stop margin ϵ	10^{-3}
Class readout	Sample from softmax($\hat{z}$) at $t = 1 - \epsilon$

E.2.2 Extension to Multimodal Meta Flow Maps.

The multimodal Meta Flow Map uses the same DiT backbone as the unconditional multimodal flow model, with three additional conditioning inputs: the inner times (s, u) and the outer state $x_{\mathrm{cond}} = x_t = (y_t, a_t)$. The current inner state $\bar{x}_s = (\bar{y}_s, \bar{a}_s)$ is embedded as in the base model,

$$h_{\bar{y}} = W_y \bar{y}_s + e_y, \qquad h_{\bar{a}} = W_a \bar{a}_s + e_a.$$

The outer conditioning state is embedded by a separate conditional projection,

$$h_y^{\mathrm{cond}} = W_y^{\mathrm{cond}} y_t + e_y^{\mathrm{cond}}, \qquad h_a^{\mathrm{cond}} = W_a^{\mathrm{cond}} a_t + e_a^{\mathrm{cond}}.$$

We inject this conditioning additively into the corresponding modality tokens, gated by an embedding of the outer time t

$$h_y^{(0)} = h_{\bar{y}} + g_y(t) \odot h_y^{\mathrm{cond}}, \qquad h_a^{(0)} = h_{\bar{a}} + g_a(t) \odot h_a^{\mathrm{cond}}.$$

The gates $g_y(t), g_a(t)$ are produced by an MLP from the Fourier embedding of t and are initialized so that the conditional branch vanishes at the unconditional limit.

The inner Meta Flow Map times are encoded separately,

$$c_{s,u,t} = \phi_s(s) + \phi_u(u) + \phi_t(t),$$

and passed to each transformer block through adaLN-Zero modulation. Thus each block receives shift, scale, and gate parameters

$$(\beta_\ell, \gamma_\ell, \alpha_\ell) = \mathrm{MLP}_\ell(c_{s,u,t}),$$

Table 13: Paired reduction in exactly computed target error. Positive values mean that the guided sample has lower target error than its unguided pair. Parentheses give 95% confidence intervals over the paired improvement.

Reward	Length L	Paired error reduction [95% CI]
Motif	50	$0.126\,[0.109,\ 0.144]$
Motif	100	$0.106\,[0.093,\ 0.121]$
Motif	200	$0.095\,[0.084,\ 0.105]$
Motif	400	$0.066\,[0.056,\ 0.075]$
Conjunction	50	$0.092\,[0.080,\ 0.106]$
Conjunction	100	$0.067\,[0.058,\ 0.076]$
Conjunction	200	$0.057\,[0.050,\ 0.064]$
Conjunction	400	$0.044\,[0.038,\ 0.051]$

and applies

$$h \leftarrow h + \alpha_\ell \odot f_\ell((1+\gamma_\ell) \odot \mathrm{LN}(h) + \beta_\ell)\,,$$

for both attention and MLP sublayers. The output heads predict a continuous endpoint $\widehat{\Psi}^y_{s,u} \in \mathbb{R}^{d_y}$ and categorical logits $\hat{\mathbf{z}}^a_{s,u}$, with

$$\widehat{\boldsymbol{\Psi}}^a_{s,u} = \mathrm{softmax}(\hat{\mathbf{z}}^a_{s,u}).$$

The model is trained with diagonal matching and ESD consistency, using MSE for the continuous head and KL / cross-entropy for the discrete head. We trained the multiMFM for 24 hours on one 80GB NVIDIA H100 GPU.

F Experimental Details

All experiments were done on A100 or H100 NVIDIA GPUs with 80GB RAM.

F.1 Exactly evaluable DNA rewards

The cyclizability oracle remains a learned model and therefore cannot completely exclude shared predictor bias. We therefore evaluate dMFM using rewards that can be computed directly from the generated sequence. The first reward applies a smooth maximum over occurrences of the motif `TTTTTC`. The second applies a smooth minimum to the occurrence scores of `TTTTTC` and `AAAATT`, requiring both motifs.

For every reward and length $L \in \{50, 100, 200, 400\}$, we generate 100 paired guided and unguided samples. Each pair uses identical initial noise, and all guidance hyperparameters are fixed across lengths. We evaluate the exact target error, uniqueness, normalized pairwise Hamming distance, nearest distance to a parent-disjoint test set, and Evo 2-7B negative log-likelihood.

Data splitting was done as follows. 230 parents for the generator and guide, 230 for the oracle, 29 + 29 validation parents, and 58 held-out test parents. Windows are extracted after splitting, giving $\lfloor (1044 - L)/7 \rfloor + 1$ windows per parent (143, 135, 121, 93 for $L = 50, 100, 200, 400$), so every split is an exact whole-parent multiple. All generated sequences remain unique, and the changes in Hamming diversity and Evo 2 likelihood are small or favorable.

Guidance improves 93–98% of matched pairs across the eight settings. Every generated sample is unique. Relative to unguided sampling, normalized pairwise Hamming diversity decreases by only 0.008–0.020, while nearest-test distance decreases by 0.003–0.012. Evo 2 NLL improves in every setting. Thus, target attainment persists under exactly computed rewards without evidence of sample duplication or reduced sequence-model plausibility.

F.2 Scale-free Monte Carlo protocol

Let $g^\star$ be an independent GLASS estimate using 2,048 terminal-noise samples, and let $\widehat{g}_N$ be an estimate using N independent terminal samples from the dMFM sampler. For S conditioning states,

we define

$$\mathrm{RelL1}(\widehat{g}_N, g^\star) = \frac{\sum_{s=1}^{S} \|\widehat{g}_{N,s} - g_s^\star\|_1}{\sum_{s=1}^{S} \|g_s^\star\|_1}, \tag{46}$$

$$\mathrm{Cos}(\widehat{g}_N, g^\star) = \frac{1}{S} \sum_{s=1}^{S} \frac{\langle \widehat{g}_{N,s}, g_s^\star \rangle}{\|\widehat{g}_{N,s}\|_2 \|g_s^\star\|_2}. \tag{47}$$

The pooled normalization avoids unstable ratios for individual states whose reference gradient is nearly zero.

For every length, we use 32 held-out conditioning states and eight independent noise pools per state. Table 14 reports representative Monte Carlo budgets; the complete sweep from $N = 1$ to $N = 256$ appears in Appendix F.3.

Table 14: Scale-free finite-MC gradient accuracy. Each entry is relative ℓ_1 error / cosine similarity against an independent GLASS-2048 reference. Lower relative error and higher cosine are better.

Reward	MC samples	50 bp	100 bp	200 bp	400 bp
GC	8	1.92/.24	1.67/.27	1.40/.32	1.48/.26
GC	128	.82/.64	.72/.69	.58/.77	.66/.67
Motif	8	2.07/.24	1.92/.26	1.94/.26	1.47/.28
Motif	128	1.05/.57	.92/.57	.86/.61	.71/.66
Conjunction	8	2.55/.20	2.30/.21	2.19/.20	1.93/.18
Conjunction	128	1.23/.49	1.15/.50	.98/.54	.93/.52

F.3 Complete scale-free Monte Carlo results

Table 15: Complete scale-free gradient-estimation results. Each entry is relative ℓ_1 error / cosine similarity against the independent GLASS-2048 reference.

Reward	N	50 bp	100 bp	200 bp	400 bp
GC	1	3.41/.14	3.51/.15	2.71/.16	2.95/.14
GC	2	3.46/.16	2.63/.19	2.17/.19	2.45/.17
GC	4	2.19/.19	2.02/.21	1.65/.24	1.80/.22
GC	8	1.92/.24	1.67/.27	1.40/.32	1.48/.26
GC	16	1.62/.32	1.42/.36	1.11/.44	1.27/.34
GC	32	1.30/.43	1.13/.48	.94/.57	1.00/.45
GC	64	1.06/.54	.92/.58	.76/.67	.82/.57
GC	128	.82/.64	.72/.69	.58/.77	.66/.67
GC	256	.63/.74	.55/.78	.45/.85	.51/.76
Motif	1	3.18/.14	4.64/.13	4.53/.15	3.61/.15
Motif	2	3.16/.17	3.44/.15	3.14/.18	2.66/.18
Motif	4	2.67/.20	2.79/.19	2.52/.21	2.30/.22
Motif	8	2.07/.24	1.92/.26	1.94/.26	1.47/.28
Motif	16	1.93/.30	1.53/.32	1.56/.33	1.25/.36
Motif	32	1.47/.37	1.30/.37	1.33/.43	1.02/.47
Motif	64	1.23/.47	1.06/.48	1.06/.52	.88/.55
Motif	128	1.05/.57	.92/.57	.86/.61	.71/.66
Motif	256	.84/.66	.75/.67	.72/.72	.56/.76
Conjunction	1	4.18/.10	5.51/.10	5.08/.11	4.56/.10
Conjunction	2	4.03/.13	4.53/.12	3.62/.12	3.32/.12
Conjunction	4	3.33/.14	3.14/.14	2.74/.15	2.79/.16
Conjunction	8	2.55/.20	2.30/.21	2.19/.20	1.93/.18
Conjunction	16	2.22/.22	1.90/.24	1.79/.26	1.59/.25
Conjunction	32	1.72/.29	1.66/.31	1.49/.35	1.34/.32
Conjunction	64	1.49/.38	1.34/.41	1.24/.44	1.16/.41
Conjunction	128	1.23/.49	1.15/.50	.98/.54	.93/.52
Conjunction	256	1.01/.58	.92/.61	.81/.65	.76/.63

F.4 DNA Mechanics Steering

At inference-time we steer with the reward defined as

$$r(x) = -|f_{\text{cyc}}(x) - y^\star|^2 \tag{48}$$

where $y^\star$ is a target cyclyzability and f_{cyc} is a cyclyzability predictor (CNN) taking in a discrete sequence. To make it differentiable with respect to probabilities (a point on the simplex), we removed the initial one first one-hot encoded layer, so that f_{cyc} is well-defined over a superposition of discrete states.

We verified the validity of our generative backbone by distribution comparison of both cyclyzability and correlation metrics, as in Figure 6. One can further inspect the effect of tilting the dMFM in Figure 7 as a function of guidance strength, and an example of a guided sample in Figure 8.

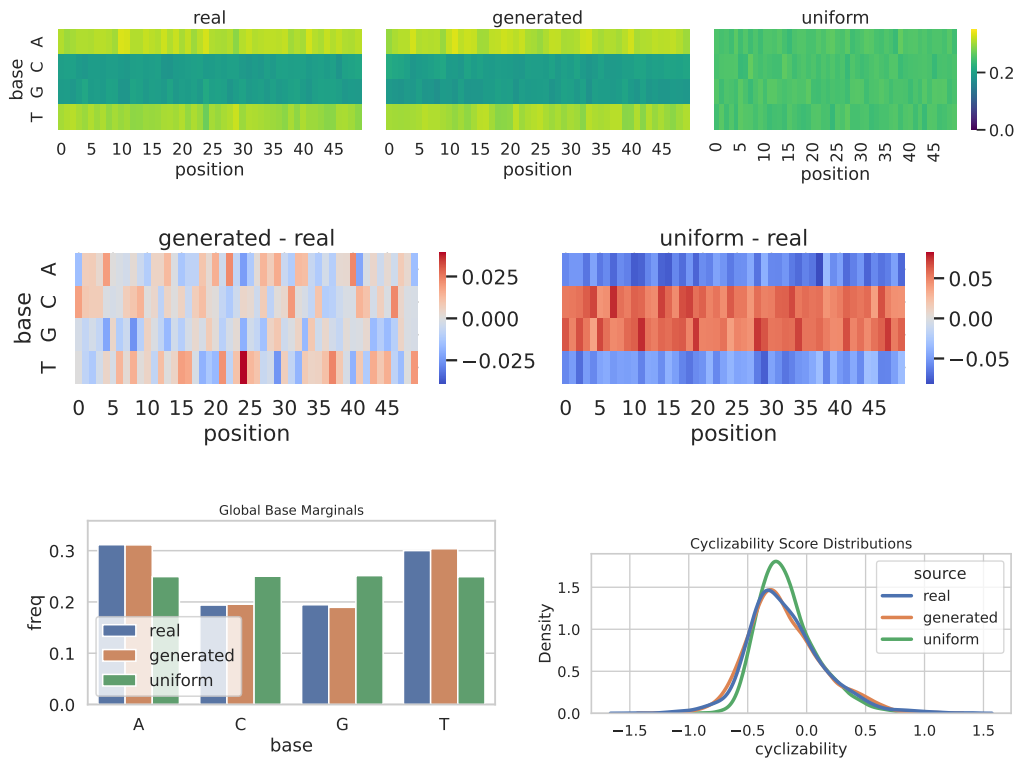
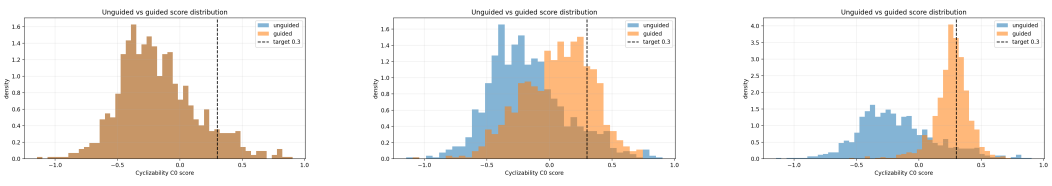


Figure 6: Comparison of real yeast DNA sequences, generated sequences, and a uniform baseline. The generated samples reproduce the real position-wise base-frequency structure and global base marginals, whereas the uniform baseline lacks the observed nucleotide biases. Cyclizability score distributions further show that generated sequences better match the real sequence distribution.



(a) No guidance. (b) Moderate guidance. (c) Strong guidance.

Figure 7: Degrees of tilt of cyclyzability distribution as strength of tilt varies.

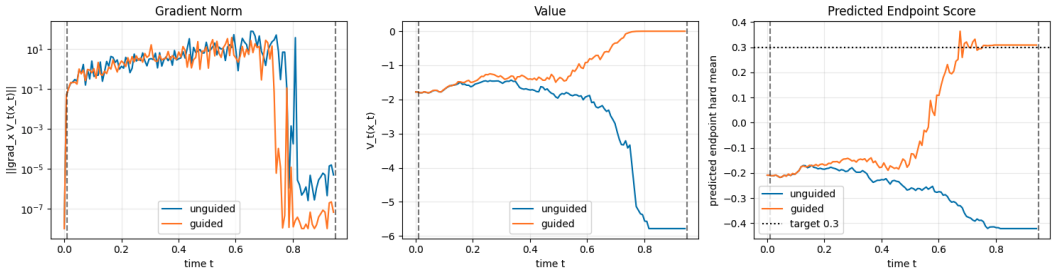


Figure 8: Demonstration of guidance using dMFM for one DNA generation sample. We compare a guided and an unguided sample. The gradient clearly push the discrete variables in ambient space towards a sequence of higher reward.

Comparison	Pearson r	p-value
Global: generated vs. real	0.9991	9.5×10^{-4}
Global: uniform vs. real	0.1813	0.819
Position-wise: generated vs. real	0.9863	6.02×10^{-157}
Position-wise: uniform vs. real	0.0264	0.711

Table 16: Pearson correlations between real yeast DNA statistics and either generated or uniform baseline sequences. The generative model closely matches both global and position-wise base frequencies, while the uniform baseline shows little correlation with the real data.

F.5 Posterior diversity diagnostics

We ran additional DNA ablations to check posterior recovery, gradient-estimator scaling, and sequence quality after steering. Posterior recovery is evaluated by comparing normalized pairwise Hamming distances from dMFM posterior samples to the corresponding GLASS posterior sampler across sequence lengths and intermediate times. The distances closely match the GLASS reference across lengths $L \in \{50, 100, 200, 400\}$ and times $t \in \{0, 0.25, 0.5, 0.75\}$, showing that dMFM preserves posterior diversity rather than collapsing to a single endpoint under scaling. Note that it's expected that diversity will collapse for increasing t.

Table 17: Posterior diversity scaling; entries are dMFM / GLASS normalized pairwise Hamming distances.

t	$L = 50$	$L = 100$	$L = 200$	$L = 400$
0.00	0.732 / 0.736	0.737 / 0.738	0.742 / 0.740	0.731 / 0.733
0.25	0.713 / 0.716	0.711 / 0.714	0.717 / 0.717	0.714 / 0.713
0.50	0.563 / 0.555	0.552 / 0.563	0.556 / 0.559	0.567 / 0.555
0.75	0.063 / 0.059	0.056 / 0.060	0.061 / 0.061	0.067 / 0.060

F.6 dMFM approximation error against GLASS

As a separate diagnostic, we compare gradients obtained using the selected four-map dMFM posterior sampler against a GLASS reference using 128 posterior samples. We use 32 held-out states at outer time $t = 0.3$, eight matched noise pools per state, and a 50-step RK4 integration for the GLASS reference. Monte Carlo budgets are nested prefixes of the same terminal-noise pool.

Table 18 reports mean per-coordinate absolute error. These values measure the combined effects of finite-sample estimation and dMFM posterior approximation. Because per-coordinate MAE depends on gradient scale, this table is not used to compare scaling across sequence lengths; the scale-free analysis in Table 15 compares GLASS Monte Carlo error across lengths, without measuring dMFM approximation error.

Table 18: Raw per-coordinate gradient MAE of dMFM against a GLASS-128 reference. This diagnostic is scale dependent and is not used to infer how the required Monte Carlo budget changes with sequence length.

Reward	L	MC=1	MC=2	MC=4	MC=8
GC	50	0.2646	0.1472	0.1256	0.1049
GC	100	0.1468	0.1234	0.0998	0.0872
GC	200	0.1045	0.0811	0.0652	0.0616
GC	400	0.0811	0.0603	0.0507	0.0399
Motif	50	0.2188	0.2149	0.1767	0.1196
Motif	100	0.2214	0.1612	0.1125	0.0862
Motif	200	0.2224	0.1336	0.0820	0.0562
Motif	400	0.0544	0.0515	0.0460	0.0386
Conjunction	50	0.1583	0.1624	0.1434	0.1203
Conjunction	100	0.2571	0.1779	0.1477	0.1206
Conjunction	200	0.2326	0.1516	0.1062	0.0756
Conjunction	400	0.1046	0.0927	0.0691	0.0516

F.7 multiMFM Toy Example

F.7.1 Multimodal Flag Example

We first showcase an example of multiMFM in a synthetic toy example of colored particles in $\mathbb{R}^2$. The state space is $x = (c, y_1, y_2)$ where $c \in \mathcal{C} := \{\text{Blue}, \text{Red}, \text{White}\}$ and $y_1, y_2 \in \mathbb{R}$. We let $p_0 = \mathcal{N}(0, \mathbb{I}_3) \times N(0, \mathbb{I}_2)$ and $p_1 = \mathcal{U}(\mathcal{C}) \times \mathcal{U}([-1, 1]^2)$. We first train a multimodal flow matching model to learn p_1, before we train a multimodal Meta Flow Map. As our first demo we use a multimodal reward given where a reward r_{Norway} is dependent the relationship of position (continuous) and color (discrete). Here we do the steering only on the discrete coordinates, and we observe the shape of the base distribution (square) is maintained. The second reward r_{Smiley} is similarily depending on both position and color, but we here steer both position and color simultaneously. Note that as sampling is still stochastic, so we expect low-reward samples to also occur. More details are in Appendix E.2.

Throughout, let $X_1 = (c, y)$ denote the joint endpoint, where $c \in \mathbb{R}^K$ are the model-order color coordinates and $y = (y_1, y_2) \in \mathbb{R}^2$ is the spatial coordinate. We use the smooth box indicator

$$\sigma_k(z; a, b) = \text{sigmoid}\big(k(z - a)\big)\,\text{sigmoid}\big(k(b - z)\big),$$

which approximates $\mathbf{1}_{[a,b]}(z)$ as $k \to \infty$. We also use the soft union

$$a \sqcup b = 1 - (1 - a)(1 - b).$$

F.7.2 Norway flag reward

The Norway reward defines three soft regions in the (y_1, y_2) plane and assigns a target color to each region. Let x_0 denote the horizontal center of the vertical bar, and let w_W and w_B denote the white and blue half-widths, with $w_B < w_W$. The white and blue Nordic crosses are

$$W(y) = \sigma_k(y_1; x_0 - w_W, x_0 + w_W) \sqcup \sigma_k(y_2; -w_W, w_W),$$

$$B(y) = \sigma_k(y_1; x_0 - w_B, x_0 + w_B) \sqcup \sigma_k(y_2; -w_B, w_B).$$

These define nonnegative soft region scores in the color order (R, B, W):

$$s_R(y) = 1 - W(y), \qquad s_B(y) = B(y), \qquad s_W(y) = W(y)\big(1 - B(y)\big).$$

We then reorder these scores into the model color convention (B, W, R):

$$s(y) = \big(s_B(y), s_W(y), s_R(y)\big) \in \mathbb{R}^3_{\geq 0}.$$

If c is represented as probabilities, the reward is

$$r_{\text{Norway}}(X_1) = \langle c, s(y) \rangle.$$

The log-reward used for value estimation is

$$\log r_{\text{Norway}}(X_1) = \log(\max\{\langle c, s(y)\rangle, \varepsilon\}) .$$

If c is represented as logits, we instead use the region-conditioned log-softmax reward

$$\log r^{\text{soft}}_{\text{Norway}}(X_1) = \langle s(y), \text{logsoftmax}(c)\rangle .$$

Optionally, $s(y)$ is stop-gradiented when we want the reward to guide only the categorical coordinates.

F.7.3 SMILEY REWARD

The smiley reward replaces the cross-shaped regions with three localized features: two Gaussian eyes and a parabolic mouth. Define

$$G(y; \mu, \sigma) = \exp\Big(-\frac{\|y-\mu\|_2^2}{2\sigma^2}\Big) .$$

With eye centers

$$\mu_R = (-x_e, y_e), \qquad \mu_B = (x_e, y_e),$$

and shared scale σ_e, define

$$E_R(y) = G(y; \mu_R, \sigma_e), \qquad E_B(y) = G(y; \mu_B, \sigma_e).$$

The mouth is a Gaussian tube around the parabola $y_2 = y_0 + \kappa y_1^2$, gated to a finite rectangle

$$M(y) = \exp\Big(-\frac{(y_2 - y_0 - \kappa y_1^2)^2}{2\sigma_m^2}\Big)\, \sigma_k(y_1; -m_x, m_x)\sigma_k(y_2; m_y^-, m_y^+).$$

A square gate restricts the reward to the learned spatial support:

$$S(y) = \sigma_k(y_1; -1, 1)\sigma_k(y_2; -1, 1).$$

In model color order (B, W, R), the feature scores are

$$s(y) = S(y)\big(E_B(y), M(y), E_R(y)\big) \in \mathbb{R}^3_{\geq 0}.$$

These scores are not normalized; the smiley features are localized, so most of the plane has $s(y) \approx 0$.

The smiley reward couples spatial features to color probabilities by

$$r_{\text{Smiley}}(X_1) = \sum_{k=1}^{K} \text{softmax}(c/T)_k \, s_k(y),$$

where T is the color temperature. The corresponding log-reward is computed as

$$\log r_{\text{Smiley}}(X_1) = \text{logsumexp}_k \left[\log\big(\max\{s_k(y), \varepsilon\}\big) + \log \text{softmax}(c/T)_k\right] .$$

This rewards samples that place the correct color on the corresponding spatial feature.

F.8 TRAINING DETAILS FOR MOLECULAR MODELS

We trained unconditional TABASCO-based molecular generators on QM9 and GEOM-Drugs, jointly modeling heavy-atom identities and three-dimensional coordinates. For each dataset, we froze the generator and trained a multimodal meta flow map (MFM) using its GLASS posterior velocity as the teacher. The same frozen QM9 MFM was used for all six property-guidance tasks. Base-model hyperparameters are listed in Table 19.

Use during guidance. The selected QM9 MFM was evaluated with four diagonal integration steps and 32 posterior samples within a 128-step outer ODE trajectory. The selected GEOM-Drugs MFM used 10 flow-map integration steps and four posterior samples within a 100-step outer ODE trajectory. These inference settings are separate from the training budgets above. These molecular experiments therefore use few-step posterior sampling rather than a single map evaluation.

Table 19: Base-model training hyperparameters for the TABASCO molecular experiments. Checkpoint steps identify the generators used for MFM distillation and guidance. The GEOM-Drugs effective batch includes seven additional random rotations of each molecule.

Hyperparameter	**QM9**	**GEOM-Drugs**
Data / state		
Training set	50,000 molecules	$\approx$ 1.17M conformers
Hydrogen atoms	Excluded	Excluded
Maximum heavy atoms in data	9	71
Atom-vector dimension	9	9
Coordinate normalization divisor	1.0	2.0
Probability path	Linear	Linear
Training-time distribution	$\mathrm{Beta}(1.8, 1)$	$\mathrm{Beta}(1.8, 1)$
Architecture		
Hidden width	64	256
Transformer layers	4	16
Attention heads	4	8
Cross-attention	Enabled	Enabled
Optimization		
Optimizer	AdamW	Adam
Optimizer (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.995)$
Learning rate	2×10^{-3}	2×10^{-3}
Learning-rate schedule	Constant	Constant
Weight decay	0.01	0
Gradient clipping (ℓ_2)	1.0	0.5
EMA decay	—	0.999
Training		
Batch before rotation copies	128	512
Additional rotation copies	0	7
Batch including rotation copies	128	4,096
Selected checkpoint step	100,000	89,310
Coordinate / atom loss coefficients	1.0 / 0.1	1.0 / 0.1

F.9 Mode collapse checks on molecular steering

PoseBusters (PB) validity is reported as a percentage of all generated molecules, including conversion failures in the denominator. QM9 used $1{,}000$ molecules per property; the six evaluations share the same unsteered baseline. The GEOM-Drugs evaluation generated $4{,}096$ molecules; its reported Tanimoto similarity and PB validity come from the same run.

Each dataset was evaluated using its own experimental PB configuration.

QM9 PB validity was 98.7% before steering, 90.3–92.3% with multiMFM, and 95.8–97.8% with multiMFM-SS. For the selected GEOM-Drugs fingerprint-guidance run, validity was 87.5%. QM9 summaries contain both a uniqueness fraction and a fraction normalized by the total number generated. Let n_{SMILES} be the number of molecules for which a SMILES string was obtained, n_{unique} the number of distinct strings, and N the number generated. The recorded metrics are

$$U = \frac{n_{\mathrm{unique}}}{n_{\mathrm{SMILES}}}, \qquad U_{\mathrm{all}} = \frac{n_{\mathrm{unique}}}{N}. \tag{49}$$

SMILES were produced using RDKit's `MolToSmiles`. The uniqueness calculation precedes PB filtering: these numbers describe the molecules with generated SMILES, not the subset that passes every PB check. As seen in Table 22, the uniqueness improves under steering.

G Licences

We used QM9 for molecular property-guided generation experiments. QM9 is cited through its original dataset paper (Ramakrishnan et al., 2014); the dataset card we used lists the license as CC0 1.0. GEOM-DRUGS is taken from Axelrod and Gómez-Bombarelli (2022).

Table 20: Hyperparameters of the selected final MFM training phases. The update counts refer to these phases only. Both students were initialized from previously trained MFM checkpoints.

Hyperparameter	**QM9**	**GEOM-Drugs**
Architecture / training		
Hidden width / layers / heads	64 / 4 / 4	256 / 16 / 8
Conditioning	Additive	Additive
Velocity parameterization	Endpoint	Endpoint
Batch per GPU	4	96
Number of GPUs	1	4
Global batch	4	384
Selected updates in phase	1,000	10,000
Cosine schedule horizon	4,000	30,000
Optimization		
Optimizer	Adam	Adam
Peak learning rate	10^{-6}	10^{-4}
Warmup steps	200	500
Gradient clipping (ℓ_2)	1.0	1.0
EMA decay	0.999	0.995
Objective / conditioning		
Objective	Diagonal + ESD	Diagonal + ESD
Coordinate / atom weights	8.0 / 0.5	1.0 / 1.0
Maximum training time	0.95	0.95
Conditioning states	Teacher-generated	Noised data endpoints
Extra mass at $t_{\mathrm{cond}} = 0$	0	0.1

Table 21: PoseBusters validity before and after MFM guidance.

Dataset	**Guidance target**	**N**	**Unsteered (%)**	**MultiMFM (%)**	**MultiMFM-SS (%)**
QM9	C_v	1,000	98.7	90.3	97.4
QM9	μ	1,000	98.7	91.6	96.0
QM9	α	1,000	98.7	91.3	97.5
QM9	$\Delta\varepsilon$	1,000	98.7	90.9	95.8
QM9	$\varepsilon_{\mathrm{HOMO}}$	1,000	98.7	92.3	97.8
QM9	$\varepsilon_{\mathrm{LUMO}}$	1,000	98.7	92.2	97.8
GEOM-Drugs	Structural fingerprint	4,096	92.0	87.5	–

We used PoseBusters (Buttenschoen et al., 2024) for physical-validity evaluation; the PoseBusters repository is licensed under the MIT License.

For DNA experiments, we used 50-bp yeast DNA sequences and cyclizability measurements from the loop-seq study of Basu et al. (2021). The corresponding Nature data-availability statement reports that sequencing data are deposited in NCBI SRA under accession PRJNA667271, that source data are provided with the paper, and that there are no restrictions on data availability. We did not identify a separate software-style license for the supplementary loop-seq data.

Table 22: QM9 uniqueness for the same runs as Table 21. All percentages are rounded to one decimal place. The count column gives the exact steered numerator and denominator for U.

Target	**Unsteered U (%)**	**Steered U (%)**	**Steered counts**	**Steered U_{all} (%)**
C_v	67.0	95.6	945/989	94.5
μ	67.0	88.2	873/990	87.3
α	67.0	94.3	935/992	93.5
$\Delta\varepsilon$	67.0	87.6	865/987	86.5
$\varepsilon_{\mathrm{HOMO}}$	67.0	91.1	902/990	90.2
$\varepsilon_{\mathrm{LUMO}}$	67.0	88.8	880/991	88.0