# Active Flow Expansion for Out-of-Distribution Discovery: from Theory to Molecules

**Riccardo De Santi**
ETH Zurich
ETH AI Center
rdesanti@ethz.ch

**Bruce Lee**
ETH Zurich
ETH AI Center
bruce.lee@ai.ethz.ch

**Cristian Perez Jensen**
ETH Zurich
cjense@ethz.ch

**Kimon Protopapas**
ETH Zurich
kprotopapas@ethz.ch

**Sophia Tang**
University of Pennsylvania
sophtang@seas.upenn.edu

**Cheng-Hao Liu**
Caltech
FutureHouse
chl@caltech.edu

**Pranam Chatterjee**
University of Pennsylvania
pranam@seas.upenn.edu

**Yisong Yue**
Caltech
yyue@caltech.edu

**Andreas Krause**
ETH Zurich
ETH AI Center
krausea@ethz.ch

https://github.com/riccardodesanti/active_flow_expansion

## Abstract

Standard flow and diffusion pre-training matches the distribution of available data (e.g., molecules), which often covers only a small fraction of the valid design space. In generative discovery, however, one aims to sample valid new-to-nature designs, assigned negligible probability under, and thus inaccessible to, standard models fitted to the observed data. To overcome this limitation, we depart from data distribution matching and view a generative model through its *generable set*: the region it covers with non-negligible probability. This allows to introduce a new learning principle for *out-of-distribution flow modeling*: enlarging a model's generable set to increase coverage of the valid design space. We propose *Active Flow Expansion (*ACTFLOW*)*, a continued pre-training method that employs verifier feedback to expand a pre-trained model over new valid regions by iteratively adapting to synthetic data generated through active exploration in the learned flow representation. Theoretically, we establish to our knowledge first-of-their-kind statistical learning guarantees for out-of-distribution flow modeling, analyzing generable set expansion as a local-to-global reachability process over a learned representation. Empirically, we assess ACTFLOW with suitable out-of-distribution generative modeling metrics across small organic molecules, mid-sized drug-like molecules, therapeutic peptides, and protein sequence design tasks. Results show that ACTFLOW expands valid coverage far beyond the region modeled by the initial pre-trained model, significantly outperforming widely adopted synthetic flow pre-training methods.

## 1 Introduction

Large-scale generative modeling has advanced rapidly in recent years, with flow [32, 33] and diffusion models [47, 48, 25] emerging as powerful approaches for generating high-fidelity samples

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

| Method | GEOM-Drugs Molecules | | Therapeutic Peptides | | Protein Sequences | |
|---|---|---|---|---|---|---|
| | Coverage ↑ | Diversity ↑ | Coverage ↑ | Diversity ↑ | Coverage ↑ | Diversity ↑ |
| Pre-trained | 35.89 ± 2.05 | 255.03 ± 1.09 | 44.33 ± 3.46 | 13.45 ± 0.14 | 66.50 ± 5.63 | 12.87 ± 0.63 |
| REC-NF | 44.67 ± 2.36 | 267.10 ± 2.88 | 0.00 ± 0.00 | 0.00 ± 0.00 | 63.75 ± 11.45 | 11.85 ± 0.34 |
| REC-F | 89.33 ± 11.56 | 284.30 ± 2.62 | 59.67 ± 17.97 | 13.62 ± 4.34 | 49.50 ± 9.60 | 11.67 ± 0.40 |
| ACTFLOW | **144.30** ± 19.28 | **303.10** ± 5.71 | **358.33** ± 95.45 | **58.87** ± 25.98 | **102.75** ± 18.36 | **42.14** ± 10.85 |

Table 1: ACTFLOW expands model coverage (i.e., number of valid clusters) and diversity (i.e., Vendi score [19] estimated via valid samples) across domains from de novo 3D molecular design to protein sequence design, significantly outperforming widely adopted recursive self-generation baselines.

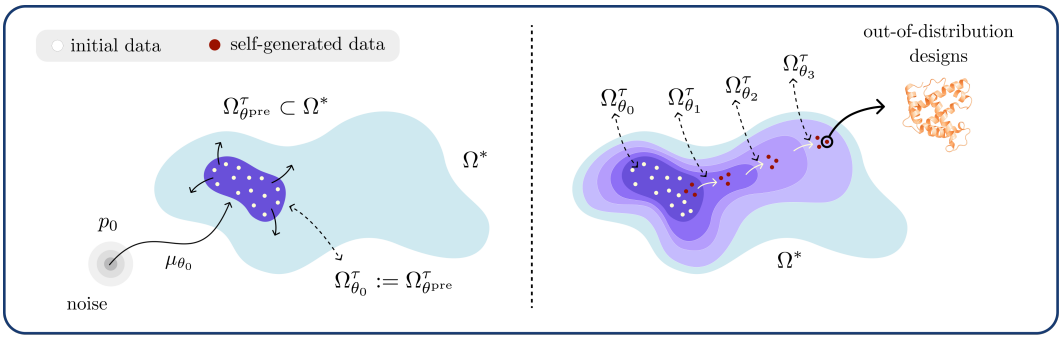



Figure 1: (left) Pre-trained flow model $\mu_{\theta_0}$ generates with sufficient probability a set $\Omega^{\tau}_{\theta_0}$ poorly covering the valid design space $\Omega^*$, i.e., $\Omega^{\tau}_{\theta_0} \subset \Omega^*$. (right) Active Flow Expansion (ACTFLOW) increases model coverage over new valid regions of $\Omega^*$, enabling out-of-distribution flow modeling.

across domains including chemistry [26], biology [12], and robotics [11]. Alongside this progress, over the last years, a growing literature has introduced test-time reward-tilting techniques [e.g., 55, 40, 54, 14]. These advances have expanded the use of generative models in scientific discovery, enabling applications such as molecular property optimization [22], or functional enzyme design [42].

**A Core Limitation of Standard Flow Pre-training for Discovery.** Despite recent progress, generative discovery remains largely constrained by standard pre-training. Flow and diffusion models are learned by matching the distribution of available data [49, 33], which typically covers only a limited portion of the valid design space. This objective is natural when the goal is to reproduce observed examples, e.g., in robotics for manufacturing, where behavioral cloning is often sufficient [39], but it is poorly suited to out-of-distribution discovery, where one typically seeks valid new-to-nature designs with negligible probability under the data distribution, and arbitrarily far from available data acquired via nature's evolution and prior discoveries. In other words, the goal of out-of-distribution discovery is in tension with standard generative modeling as distribution matching. Nonetheless, pre-trained generative models remain, to date, arguably the most promising way to access the complex design spaces in which scientific discovery takes place. This raises the central question of this work:

*How can we adapt a pre-trained flow or diffusion model in a task-agnostic way to enable out-of-distribution discovery?*

Answering this question would advance the algorithmic-theoretical foundations of generative discovery, with direct implications for high-impact scientific discovery applications. We take a step toward this goal by moving beyond data distribution matching and *rethinking flow model learning* for out-of-distribution generative modeling. Concretely, we make the following contributions.

**Our contributions.**

- We introduce the notion of *generable set*: the region of design space a model can access with non-negligible probability. This yields a rigorous mathematical framework for *out-of-distribution flow modeling*, and a new learning principle for continued pre-training: expanding a model generable set to cover a larger portion of the valid design space, which we name *generable set expansion*.
- We propose Active Flow Expansion (ACTFLOW), a continued pre-training method that expands a pre-trained flow model through recurrent adaptation on self-generated synthetic data – as illustrated in Fig. 1 (right). ACTFLOW actively explores in the *learned diffusion representation* at intermediate noise levels, where the geometry is more favorable to global exploration.

- We establish, to our knowledge, first-of-their-kind statistical learning guarantees for out-of-distribution flow modeling. These rely on an energy-based modeling abstraction, and analyze active generable set expansion as a local-to-global reachability process in a learned representation.
- We employ domain-agnostic metrics for out-of-distribution generative modeling and assess ACTFLOW across small organic molecules, mid-sized drug-like molecules, therapeutic peptides, and protein design. ACTFLOW significantly increases valid coverage beyond the initial pre-trained model, vastly outperforming widely used recursive data generation baselines. We further show that the same algorithmic principles and empirical gains extend to discrete (diffusion) models.

## 2 From Data Distribution Matching to Out-of-Distribution Flow Modeling

We consider a design space $\mathcal{X} \subseteq \mathbb{R}^d$, where each $x \in \mathcal{X}$ is a candidate design, such as a molecule or protein. Let $p_{data}$ denote the distribution of the available data, assumed to consist of i.i.d. samples that satisfy the domain-specific notion of validity, e.g., valid molecules. A continuous-time flow model is defined by a time-dependent velocity field $u_\theta : \mathcal{X} \times [0,1] \to \mathcal{X}$ and the ordinary differential equation

$$\frac{dx_t}{dt} = u_\theta(x_t, t), \qquad x_0 \sim p_0, \tag{1}$$

where $p_0$ is a simple base distribution, typically Gaussian [33]. Let $p_t^\theta$ denote the marginal density induced at time $t$ by the flow, and let $p_1^\theta$ denote its terminal density. In standard flow matching pre-training, one specifies conditional interpolation paths $p_t(\cdot \mid x_0, x_1)$ between $x_0 \sim p_0$ and $x_1 \sim p_{data}$, with associated target velocity field $u_t^\star(\cdot \mid x_0, x_1)$, and learns $u_\theta$ by minimizing the loss:

$$\mathbb{E}\left[\|u_\theta(x_t, t) - u_t^\star(x_t \mid x_0, x_1)\|_2^2\right], \qquad t \sim U[0,1],\ x_t \sim p_t(\cdot \mid x_0, x_1). \tag{2}$$

This vector-field regression objective allows to match the terminal distribution $p_1^\theta$ to the available-data distribution $p_{data}$, as by Eq. 3. Beyond flow matching, the same distribution-matching learning principle underlies also standard pre-training of continuous [49] and discrete [35] diffusion models.

**Distribution Matching: Standard Flow Model Pre-Training**

$$\text{learn } \theta \text{ such that } p_1^\theta \approx p_{data}. \tag{3}$$

### 2.1 Generable Set Expansion: Flow Modeling Beyond Data Distribution Matching

In this work, we depart from viewing generative models as data distribution approximators and instead cast them as *valid design space approximators*. Let $v : \mathcal{X} \to \{0,1\}$ denote whether a design is valid according to the domain one wishes to model, e.g., $v$ might express molecular physiochemical validity, protein sequence foldability, etc. We indicate via $\Omega^*$ the entire valid design space:

$$\Omega^* := \{x \in \mathcal{X} : v(x) = 1\}, \tag{4}$$

which we wish our generative model to cover well. To capture the data regions sufficiently covered by a pre-trained flow model $\mu_\theta$, we introduce the following notion of *generable set*.

**Definition 1** (Generable Set)**.** *Let $\mu_\theta$ be a flow model inducing terminal density $p_1^\theta$ on $\mathcal{X}$. For $\tau > 0$, we define its $\tau$-level* generable set *as*

$$\Omega_\theta^\tau := \{x \in \mathcal{X} \mid p_1^\theta(x) \geq \tau\}. \tag{5}$$

For $\tau \to 0$ the generable set approximates the model support, i.e., $\Omega_\theta^0 := \mathrm{Supp}(p_1^\theta)$. However, the support can include regions assigned only infinitesimal, yet positive probability, and therefore may substantially overstate the model's effective coverage. Instead, for sufficiently small $\tau > 0$, the generable set $\Omega_\pi^\tau$ captures the region likely to be sampled under a finite budget. With this notion of generable set, we can state the following limitation of standard pre-trained flows for out-of-distribution discovery [1]:

**Central Limitation of Standard Flow Pre-training**

$$\Omega_{\theta^{\mathrm{pre}}}^\tau \subset \Omega^* \qquad \text{with} \qquad \mathrm{Vol}(\Omega_{\theta^{\mathrm{pre}}}^\tau) \ll \mathrm{Vol}(\Omega^*). \tag{6}$$

[1] For clarity, here we consider a model generating only valid designs. In general $\Omega_{\theta^{\mathrm{pre}}}^\tau \cap (\mathcal{X} \backslash \Omega^*)$ is non-empty (Apx. B)

Under exact distribution matching, Eq. (6) allows to define the valid out-of-distribution (OOD) region:

$$\text{Valid OOD region:} \quad \bar{\Omega}^{\tau}_{\theta^{\text{pre}}} := \Omega^* \setminus \Omega^{\tau}_{\theta^{\text{pre}}} \tag{7}$$

This is the set of valid designs not reliably covered by the pre-trained model $\mu_{\theta_{\text{pre}}}$ and typically not represented in $p_{\text{data}}$, such as new-to-nature designs. To overcome this limitation, we introduce *generable set expansion* (see Eq. 8), a new learning principle for continued pre-training: given a standard pre-trained flow model, expand its generable set to cover a larger portion of the valid design space $\Omega^*$.

**Generable Set Expansion: Out-of-Distribution Flow Modeling**

$$\text{learn } \theta \text{ such that } \Omega^{\tau}_{\theta} \approx \Omega^*. \tag{8}$$

### 2.2 Generable Set Expansion via Iterative Continued Pre-training

Given a pre-trained flow model $\mu_{\theta^{\text{pre}}}$, we approach generable set expansion through continued pre-training on self-generated data. Beyond the data distribution, validity must be obtained through black-box verifier queries, e.g., molecular validity [13, 21] or protein foldability [27, 57]. Since such queries can only be made on generable designs, expansion is intrinsically recursive, as shown in Fig. 1:

$$\theta^{\text{pre}} = \theta_0 \to \theta_1 \to \cdots \to \theta_T, \qquad \Omega^{\tau}_{\theta_0} \subseteq \Omega^{\tau}_{\theta_1} \subseteq \cdots \subseteq \Omega^{\tau}_{\theta_T} \approx \Omega^{\star}.$$

The central algorithmic question is therefore how to self-generate data that expands the model's generable set, rather than merely reinforcing regions already assigned high density. Standard sampling is poorly suited to this goal: it preferentially draws from dominant modes, and may therefore neglect valid frontier regions that enable valid expansion. We report an illustrative Gaussian analysis in Apx. C.1 to isolate this failure mode in a minimal setting. The analysis shows that, even when expansion-enabling samples remain within the current generable set, passive self-generation can be exponentially unlikely to sample them when valid directions of expansion are sparse. This highlights the core algorithmic requirement: self-generation must steer sampling toward generable and valid *frontier* regions, rather than merely reinforce dominant modes. In the next section, we introduce ActFlow, which implements this principle and yields provable generable set expansion guarantees.

## 3 Algorithm: Active Flow Expansion

We now introduce Active Flow Expansion (ActFlow) (Alg. 1), a continued pre-training method for *generable set expansion*: it expands a pre-trained flow model's generable set to cover a larger portion of the valid design space. Concretely, ActFlow fine-tunes the model on self-generated data acquired via inference-time active exploration in a learned flow representation at intermediate noising levels.

**High-level Algorithm Summary.** At round $t$, ActFlow fits a verifier uncertainty estimate $\sigma_t(\cdot)$ from a buffer $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{t}$, self-generates a new candidate, or a batch, $x_{t+1}$ via Eq. (9), queries the verifier to obtain $y_{t+1} = \tilde{v}(x_{t+1})$, appends $(x_{t+1}, y_{t+1})$ to $\mathcal{D}_t$, and updates the flow into $\mu_{\theta_{t+1}}$.

**Active Exploration over a Noised Flow Representation.** ActFlow performs active self-generation in the learned representation $\mathcal{Z}_s$, given by hidden features of the velocity network $\mu_\theta$ at noising level $s \in (0, 1)$. At each iteration, Algorithm 1 steers the current model at inference-time toward high verifier-uncertainty regions in $\mathcal{Z}_s$, while remaining close to the current model density:

**(Inference-Time) Active Exploration over Noised Flow Representation $\mathcal{Z}_s$**

$$x_{t+1} \sim \tilde{p}_t \in \arg\max_{q} \mathbb{E}_{x\sim q}\left[\sigma_t\left(\phi^t_s(x)\right)\right] - \beta\, \mathrm{KL}\left(q \,\|\, p_1^{\theta_t}\right). \tag{9}$$

Here, $\phi^t_s : \mathcal{X} \to \mathcal{Z}_s$ denotes the representation extracted from the velocity network at noising level $s \in (0, 1)$. The first term of Eq. 9 favors informative queries about the verifier $v$, and thus about the valid design space $\Omega^*$. The KL term regularizes search toward the current generative prior, biasing exploration toward likely valid regions. The parameter $\beta$ controls the exploration–prior trade-off: as $\beta \to \infty$, Eq. (9) recovers standard sampling, $x_{t+1} \sim p_1^{\theta_t}$; as $\beta \to 0$, it becomes pure uncertainty

**Algorithm 1** Active Flow Expansion (ACTFLOW)

**Require:** Flow model $\mu_{\theta_0}$, black-box verifier $\tilde{v}$, representation time-step $s \in (0,1)$, iterations $T$

$\mathcal{D}_0 \leftarrow \emptyset$
**for** $t = 0, 1, \ldots, T-1$ **do**
  Update surrogate uncertainty $\sigma_t$ from $\mathcal{D}_t$
  Self-generate:

$$x_{t+1} \sim \tilde{p}_t \in \arg\max_q \mathbb{E}_{x\sim q}\big[\sigma_t\big(\phi_s^t(x)\big)\big] - \beta\,\mathrm{KL}\Big(q \,\|\, p_1^{\theta_t}\Big)$$

  Query verifier: $y_{t+1} \leftarrow \tilde{v}(x_{t+1})$
  $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$
  $\theta_{t+1} \leftarrow \textsc{UpdateFlow}(\theta_t, \mathcal{D}_{t+1})$
**end for**
**return** $\mu_{\theta_T}$

maximization, which may target remote regions where the model prior might no longer provide a useful validity bias.

To instantiate Eq. (9), we model verifier uncertainty directly in $\mathcal{Z}_s$. We then view the verifier labels as noisy observations $y_t = \tilde{v}(x_t)$ of an unknown validity function over the representation space $\mathcal{Z}_s$. To represent the uncertainty about this function, we use a linear kernel over the learned representation space: $k_{\phi_s^t}(x, x') := \langle \phi_s^t(x), \phi_s^t(x') \rangle$. Then, for a set of queried designs $x_1, \ldots, x_t$, let $X_t := (x_1, \ldots, x_t)$, let $K_{t,\phi_{s,t}} \in \mathbb{R}^{t\times t}$ be the kernel matrix with entries $(K_t)_{ij} = k_{\phi_s^t}(x_i, x_j)$, and write $k_{\phi_s^t}(x, X_t) := \big(k_{\phi_s^t}(x, x_1), \ldots, k_{\phi_s^t}(x, x_t)\big)$. We then express our uncertainty about the unknown verifier via the following closed-form expression:

$$\sigma_t^2(x) = k_{\phi_s^t}(x,x) - k_{\phi_s^t}(x, X_t)\big(K_{t,\phi_s^t} + \lambda I\big)^{-1} k_{\phi_s^t}(X_t, x), \qquad \sigma_t(x) := \sqrt{\sigma_t^2(x)}. \tag{10}$$

This corresponds to the posterior variance in Bayesian linear regression under Gaussian observations.

**Model update.** The uncertainty estimator is fit on the pairs in $\mathcal{D}_t$. The flow model is then updated by replay-based continued pre-training. Let $\mathcal{D}_t^+ = \{x : (x,1) \in \mathcal{D}_t\}$ and $\mathcal{D}_t^- = \{x : (x,0) \in \mathcal{D}_t\}$ denote accepted and rejected samples, and let $U_t^\pm \subseteq \mathcal{D}_t^\pm$ be minibatches and $\widehat{L}_t^\pm(\theta)$ their respective standard flow-matching loss terms. We use the signed update $g_t = \nabla \widehat{L}_t^+(\theta_t) - \alpha_t \nabla \widehat{L}_t^-(\theta_t)$, where rejected samples are seen as an unlearning signal [1]; see Apx. D.1. In practice, our results often hold with $\alpha_t = 0$, i.e., standard flow matching [33] on verifier-accepted samples.

Across iterations, ACTFLOW reallocates model mass from dominant pre-trained modes toward newly discovered valid regions. It remains to show that this process truly expands the model's generable set, rather than only redistributing density within it. The next section answers this question affirmatively by establishing statistical guarantees for out-of-distribution generative modeling via ACTFLOW.

## 4 Statistical Guarantees for Out-of-Distribution Flow Modeling

We now theoretically analyze Active Flow Expansion (ACTFLOW), thereby providing a first guarantee for out-of-distribution generative modeling in terms of generable set expansion rather than distribution matching of $p_{\text{data}}$. Formal theorem and detailed derivations are reported in Appendix. F.

As a first step, since ACTFLOW effectively performs active exploration in the learned representation $\mathcal{Z}_s$, we introduce the following notion of *generable representation set* $\Omega^\tau_{\pi,\phi}$ for a fixed map $\phi := \phi_s$.

**Definition 2** (Generable Representation Set). *Let $\phi : \mathcal{X} \to \mathcal{Z}$ be fixed, and let $\pi$ be a diffusion model inducing design-space density $p_1^\pi$ on $\mathcal{X}$. Let $p_1^{\pi,\phi}$ denote the induced density on $\mathcal{Z}$ obtained by pushing forward $p_1^\pi$ through $\phi$. For $\tau > 0$, we define its $\tau$-level* generable representation set *as*

$$\Omega^\tau_{\pi,\phi} := \{z \in \mathcal{Z} \mid p_1^{\pi,\phi}(z) \geq \tau\}. \tag{11}$$

**Formal feedback model: logistic bandit feedback** We now formalize the verifier as a probabilistic model over $Z = \phi(X)$. At each round $t$, ACTFLOW generates a sample $z_t := \phi(x_t)$ and observes a binary label $y_t \in \{0, 1\}$, generated according to a logistic model:

$$\Pr[y_t = 1 \mid z_t] = s(g(z_t)) \tag{12}$$

where $g : Z \to \mathbb{R}$ is an unknown latent *validity score*, and $s : \mathbb{R} \to [0, 1]$ is the sigmoid function. We assume there exists a threshold $h \in [0, 1]$, such that we can formally define the valid design space as:

$$\Omega_* := \{z \in Z : s(g(z)) \geq h\}. \tag{13}$$

For the sake of analysis, we assume $g$ lies in an RKHS $\mathcal{H}_k$ with $\|g\|_k \leq B$, $\int_{\mathcal{Z}} \exp(g(z)) \leq \bar{Z}$, and is $L_g$-Lipschitz in a metric $d$ over $\mathcal{Z}$. This model allows us to construct *anytime confidence sequences* for logistic prediction [38]. Next, we model ACTFLOW's uncertainty-tilted sampling procedure.

**Uncertainty-tilted sampling oracle modeling** We model the inference-time uncertainty sampling in Eq. (9) as approximately maximizing verifier uncertainty within the current generable set $\Omega_t^\tau$ for some $\tau > 0$. Concretely, at round $t$ it returns $x_t$ such that $z_t = \phi(x_t) \in \Omega_t^\tau$ satisfies:

$$\sigma_t(z_t) \;\geq\; \frac{1}{\alpha} \max_{z \in \Omega_t^\tau} \sigma_t(z), \qquad \alpha \geq 1. \tag{14}$$

We depart from standard bandit analyses, which assume global search over $\Omega^\star$ by instead restricting the (generative) sampler to approximate uncertainty maximization over the generable set. We suppose that the $\tau$-level generable set of the pre-trained model contains a non-empty set $S_0$ of valid designs. Further regions can be reached only after intermediate expansion steps. We formalize a local expansion step via the following *one-step reachability operator over the learned representation*:

$$R_\epsilon(S) := \{z \in \mathcal{Z} : \exists z' \in S \text{ s.t. } s(g(z')) - L_s L_g d(z, z') - \epsilon \geq h\}. \tag{15}$$

Here, $R_\epsilon(S)$ contains the representations whose validity is certifiable from $S$ up to accuracy $\epsilon$. Let $R_\epsilon^H(S_0)$ denote its $H$-fold recursive application starting from $S_0$. This enables viewing of ACTFLOW as executing a local-to-global expansion process toward the reachable valid set $R_\epsilon^H(S_0)$.

Given these objects, we can now present the main theorem, stated formally in Theorem F.5, which allows to state set-theoretic guarantees for out-of-distribution generative modeling.

**(Informal) Assumptions:**

- **(Well-specified verifier).** Verifier labels follow a calibrated logistic model in the learned representation space, $\mathbb{P}(y = 1 \mid z) = s(g(z))$, where the latent validity score $g$ has bounded RKHS norm and is Lipschitz continuous in the learned representation.
- **(Approximate and local uncertainty sampling).** At each round, the sampler approximately maximizes verifier uncertainty over the current generable set $\Omega_t^\tau$, as by Eq. (14).
- **(EBM generative update).** We employ an energy-based model (EBM) abstraction, detailed in Assump. F.1, to abstract flow model endpoint density updates as learning an implicit energy functional over the flow-learned representation $Z$. This captures that flow models preserve high density on regions that our verifier regards as valid, and maintain low density on points far from this set.

**Theorem 4.1** ((Informal) Generable representation set covers reachable set)**.** *Fix $\epsilon > 0$ and an integer $H \geq 1$. Let $\tau$ and $T^*$ satisfy*

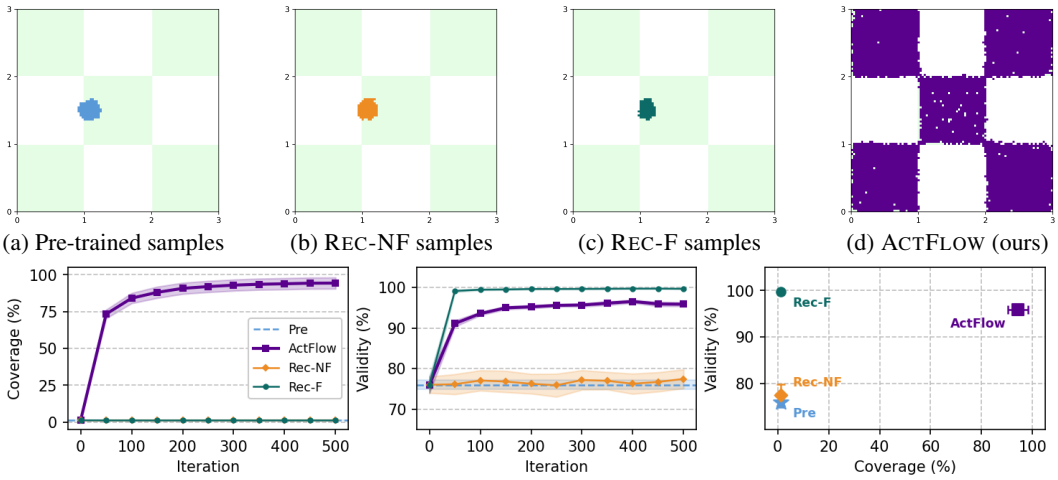
$$\tau \leq \frac{h}{(1-h)\bar{Z}} \quad \textit{and} \quad T^\star \;\gtrsim\; \left(\frac{\alpha \gamma_{HT^*}^{\Omega_\star}}{\epsilon}\right)^2, \tag{16}$$

*with up to problem-dependent constants in $\gtrsim$. Define the maximum information gain from $t$ samples by*

$$\gamma_t^A \triangleq \max_{z_1,\ldots,z_t \in A} \frac{1}{2} \log\det\left(I_t + (\lambda\kappa)^{-1} K_t\right)$$

*By running* ACTFLOW*, it holds with probability at least $1 - \delta$ that after $T^\star$ verified samples,*

$$R_\epsilon^H(S_0) \subseteq\; \Omega_{T^\star}^\tau. \tag{17}$$




(a) Pre-trained samples (b) REC-NF samples (c) REC-F samples (d) ACTFLOW (ours)

(e) Coverage (f) Validity (g) Coverage-Validity

Figure 2: (2a) The valid design space (green) and the estimated generable set ($\tau = 0.01$) of a pre-trained model (blue). (2a-2d) REC-NF and REC-F fail to expand the pre-trained model's generable set, while ACTFLOW is able to discover and expand over the entire valid design space. (2e-2g) ACTFLOW increases both model coverage (1.16% to 94.27%) and validity (76% to 95.9%), REC-NF fails at increasing both, while REC-F increases validity while even decreasing coverage (1.16% to 1.1%).

Theorem 4.1 is stated in the learned representation space $Z$. Under a map $\phi$ with measure-preserving regularity conditions stated in Assumption F.4, the same reachability guarantee transfers to the design-space generable set. For $\tau_X > 0$, define the design-space generable set at round $t$ by

$$\Omega_t^{X,\tau_X} := \{x \in \mathcal{X} : \; p_1^{\pi_t}(x) \geq \tau_X\}. \tag{18}$$

We also define the induced one-step reachability operator on $\mathcal{X}$ by

$$R_\epsilon^{X,\phi}(A) := \Big\{x \in \mathcal{X} : \exists x' \in A \text{ s.t. } s\big(g(\phi(x'))\big) - L_s L_g \, d\big(\phi(x), \phi(x')\big) - \epsilon \geq h\Big\}, \tag{19}$$

for any $A \subseteq \mathcal{X}$, and let $\big(R_\epsilon^{X,\phi}\big)^H(A)$ denote its $H$-fold iterate.

**Corollary 4.2** (Design-space coverage of the induced reachable valid set)**.** *Assume the conditions of Theorem 4.1 and Assumption F.4, with* $j_{\min} := \inf_{x \in \mathcal{X}} |\det J_\phi(x)| > 0$. *Let* $S_0^X := \phi^{-1}(S_0)$ *and* $\tau_X := j_{\min}\tau$. *Then, after the same number* $T^\star$ *of verified samples as in Theorem 4.1, with probability at least* $1 - \delta$,

$$\big(R_\epsilon^{X,\phi}\big)^H(S_0^X) \subseteq \Omega_{T^\star}^{X,\tau_X}. \tag{20}$$

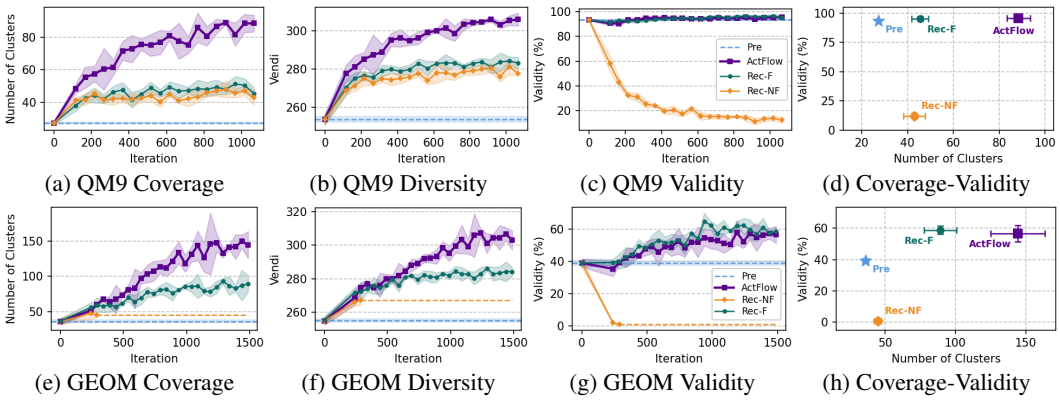
**From reachable-set to full coverage.** The guarantee is reachability-based: if $\Omega^\star$ contains components that are disconnected from $S_0^X$ in the learned geometry, local expansion provides no guarantee of covering those components. Conversely, if every $x \in \Omega^\star$ can be connected to $S_0^X$ by at most $H$ local valid expansions in representation space, then Corollary F.9 yields full valid-space coverage,

$$\Omega^\star \subseteq \Omega_{T^\star}^{X,\tau_X}.$$

Theorem 4.1 shows that ACTFLOW expands the model's $\tau$-generable set for any sufficiently small $\tau$, effectively describing finite-sample coverage. Corollary F.8 controls model validity, i.e., how tight the expanded generable set is w.r.t. $\Omega^\star$ – the opposite set-inequality for *generable set expansion*.

## 5 Experimental Evaluation of ActFlow on Molecules, Peptides, and Proteins

We evaluate ACTFLOW for expanding the valid generable set of pre-trained generative models, and compare it against widely adopted self-generation baselines: continued pre-training on unfiltered model samples (REC-NF) [46, 2] and on verifier-filtered samples (REC-F) [16, 20]. We propose two types of experiments: $(i)$ an illustrative visually interpretable setting, and $(ii)$ high-dimensional




(a) QM9 Coverage (b) QM9 Diversity (c) QM9 Validity (d) Coverage-Validity

(e) GEOM Coverage (f) GEOM Diversity (g) GEOM Validity (h) Coverage-Validity

Figure 3: (3a-3d) Molecular design on QM9 [41]. ACTFLOW expands valid coverage substantially more than REC-NF and REC-F, reaching 88.40 valid clusters versus 42.80 and 45.40 respectively (Fig. 3a). ACTFLOW also achieves the highest diversity (3b) while preserving high validity (3d). (3e-3h) Design of drug-like molecules on GEOM-Drugs [4]. ACTFLOW increases the pre-trained model coverage, diversity, and validity, vastly outperforming both REC-F and REC-NF baselines.

biochemical design tasks over molecules, therapeutic peptides, and protein sequences. Since standard generative modeling evaluation metrics are misaligned with OOD generative modeling, we employ the OOD modeling criteria presented in Apx. H.1, namely *coverage* via number of valid clusters, *diversity* via Vendi [19], and overall model validity. Additional experimental details are provided in Apx. H.8.

**Illustrative Visually Interpretable Setting.** We first evaluate ACTFLOW in a two-dimensional illustrative setting with valid design space shown in green in Fig. 2a–2d, and visualize the models' generable sets via discretization. All methods run for $T = 500$ iterations and generate $B = 64$ samples per iteration. ACTFLOW uses $\alpha_t = 0.005$, $\beta = 1/13$, and flow representation at timestep $s = 0.9$. Fig. 2a–2d show that ACTFLOW (violet) expands the pre-trained generable set ($\tau = 0.01$, Fig. 2a) to near-optimally cover the valid design space (Fig. 2c), whereas both baselines fail to expand it. Fig. 2e–2g show that ACTFLOW increases both coverage, from $1.16\%$ to $94.27\%$, and validity, from $76.00\%$ to $95.89\%$. By contrast, REC-NF leaves coverage unchanged and barely improves validity, while REC-F attains higher validity but decreases coverage. Thus, even in this simple two-dimensional setting, ACTFLOW is the only method that achieves OOD generable-set expansion, substantially improving both coverage and validity, expanding through sparse directions (corners) to new valid regions.

**Molecular design on QM9.** We evaluate ACTFLOW on FlowMol Gaussian [17], pre-trained on QM9 [41]. All methods run for 1000 iterations after 66 initial iterations without fine-tuning. ACTFLOW uses the flow representation at timestep $s = 0.9$ and $\beta = 1/10$. Fig. 3a–3d show that, relative to the pre-trained model, ACTFLOW substantially expands valid molecular coverage, reaching 88.40 valid clusters, compared to 45.40 for REC-F and 42.80 for REC-NF. ACTFLOW also achieves the highest diversity, with Vendi 306.08, while preserving high validity ($95.90\%$). In contrast, REC-F preserves validity ($95.24\%$) but expands significantly less, whereas REC-NF severely degrades validity ($12.26\%$). Thus, on QM9, ACTFLOW significantly outperforms both recursive sampling baselines, achieving a significantly stronger combination of valid coverage, diversity, and validity.

**Molecular design on GEOM-Drugs.** We further evaluate ACTFLOW on FlowMol Gaussian [17], pre-trained on GEOM-Drugs [4], a substantially larger, and more chemically relevant dataset of drug-like molecules. All methods use 2000 fine-tuning steps after a warm-up period, where 4096 samples are acquired. ACTFLOW uses flow representation timestep $s = 0.8$, $\beta = 1/7$, $\alpha_t = 0$. Fig. 3e–3h show that ACTFLOW expands valid molecular coverage substantially beyond the baselines, reaching 144.3 valid clusters, compared to 89.33 for REC-F and 44.67 for REC-NF. ACTFLOW also achieves the highest diversity, with Vendi 303.1, while maintaining comparable validity to REC-F ($56.6\%$ versus $58.83\%$). In contrast, REC-F expands substantially less, whereas REC-NF collapses in validity, as reported in Fig. 3h. Thus, on GEOM-Drugs, ACTFLOW provides a stronger expansion profile than both baselines, ultimately increasing the pre-trained model coverage by $144.3\%$ and its validity by $56.6\%$.

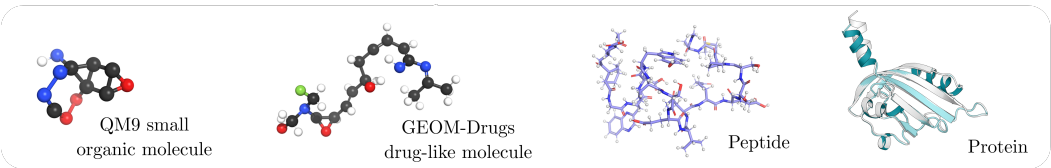



Figure 6: 3D structures directly generated or string-converted by models expanded via ACTFLOW.

| | Therapeutic Peptide Design | | | | Protein Sequence Design | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **Coverage** ↑ | **Diversity** ↑ | **FID** | **Validity** (%) ↑ | **Coverage** ↑ | **Diversity** ↑ | **FID** | **Validity** (%) ↑ |
| Pre | 44.33 ± 3.46 | 13.45 ± 0.14 | 0.00 ± 0.00 | 39.31 ± 1.02 | 66.50 ± 5.63 | 12.87 ± 0.63 | 0.05 ± 0.01 | 70.81 ± 1.12 |
| REC-NF | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.00 ± 0.00 | 63.75 ± 11.45 | 11.85 ± 0.34 | 0.27 ± 0.06 | 67.81 ± 3.11 |
| REC-F | 59.67 ± 17.97 | 13.62 ± 4.34 | 3.18 ± 1.77 | 71.16 ± 5.22 | 49.50 ± 9.60 | 11.67 ± 0.40 | 0.25 ± 0.04 | 88.12 ± 1.42 |
| ACTFLOW | 358.33 ± 95.45 | 58.87 ± 25.98 | 59.15 ± 44.31 | 41.59 ± 10.06 | 102.75 ± 18.36 | 42.14 ± 10.85 | 5.45 ± 3.34 | 83.74 ± 2.70 |

Table 2: ACTFLOW significantly expands valid model coverage (i.e., number of valid clusters) and diversity (i.e., Vendi [19]), while also increasing validity across therapeutic peptide and protein sequence design tasks. We report in red models where coverage or diversity decreased from initial model. These results show that ACTFLOW vastly outperforms widely adopted synthetic pre-training baselines.

**Therapeutic Peptide Design** We assess ACTFLOW on biological sequence generation via discrete diffusion models, as detailed in Apx. G.3. We consider the task of therapeutic peptide design [56] and employ the pre-trained SMILES discrete diffusion model from PepTune [52]. As shown in Table 2 and Fig 4, ACTFLOW significantly increases the number of clusters (i.e., coverage) computed from PeptideCLM embeddings [18] from 44.33 to 358.33, while maintaining validity, determined via the `SMILES2PEPTIDE` verifier [52]. Similarly, ACTFLOW achieves the highest diversity of 58.87 Vendi score over the same embeddings. On the contrary, REC-F maintains the similar coverage and diversity as the pre-trained model, while increasing validity, and REC-NF collapses entirely to 0.0% validity, as a single misplaced token can result in an invalid peptide – thus inducing no valid clusters. These results show that ACTFLOW's strong empirical performance extends beyond continuous flows to discrete diffusion models (see Apx. G.3), significantly increasing both model coverage and diversity, outperforming all baselines, while maintaining stable validity.

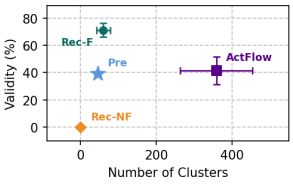



Figure 4: Peptides coverage-validity tradeoff.

**Protein Sequence Design.** We finally evaluate ACTFLOW on protein sequence design, via a continuous ESM diffusion model from SGPO [59], pre-trained on the CreiLOV fluorescence dataset [10]. We use 512 iterations with 1000 fine-tuning steps each. ACTFLOW employs flow representation timestep at $t = 0.8$. ACTFLOW substantially expands valid coverage, computed over token-level ESM embeddings, increasing the number of clusters from 66.50 to 102.75, compared to 63.75 for REC-NF and 49.50 for REC-F, which lead to significant decrease in both coverage and validity metrics, see Table 2 and Fig. 5. ACTFLOW also achieves the highest diversity, with Vendi 42.14 vs 12.87 for the pre-trained model and 11.85/11.67 for the baselines, while improving validity from 70.81% to 83.74%. In contrast, REC-F reaches higher validity (88.12%) but collapses in coverage and diversity; REC-NF reduces coverage and diversity relative to the pre-trained model. ACTFLOW achieves the strongest coverage–diversity–validity profile (see Fig. 5), significantly expanding the pre-trained protein sequence model to new valid regions of the design space.

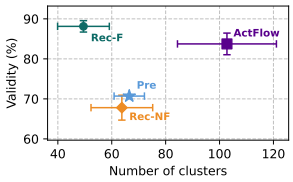



Figure 5: Proteins coverage-validity tradeoff.

## 6 Related Work

**Diffusion and flow model reward adaptation for generative optimization.** Several works adapt pre-trained diffusion and flow models for reward maximization, either via fine-tuning [e.g., 54, 14, 34] or inference-time sampling [e.g., 55, 40, 55]. Without injecting further validity information, these methods remain constrained by the pre-trained model coverage, leading to *over-optimization*, i.e.,

invalid samples, when steering the model excessively beyond the training distribution [22]. In this work, we formalize this limitation through the notion of *generable set*, and introduce ACTFLOW, a task-agnostic continued pre-training method that expands coverage to new valid regions of the design space.

**Synthetic data generation for diffusion model self-adaptation** Generative priors can provide useful synthetic data for downstream learning [e.g., 5, 36], but recursively training on generated data can induce model collapse [46, 2]. For generative model adaptation, prior work uses synthetic data for negative guidance [3] or conservative verifier-free self-play [60], with the primary goal of improving in-distribution sample quality [e.g., 3]. In contrast, we study flow adaptation for out-of-distribution modeling: rather than further refining high-density modes, the goal is to reallocate mass toward newly verified valid regions beyond the training distribution. To our knowledge, ACTFLOW is the first theory-backed synthetic continued-pretraining method for out-of-distribution generative modeling.

**Diffusion and flow based design space exploration** Recent works introduce scalable methods for diffusion- and flow-based design space exploration via entropy maximization [e.g., 15, 14] or approximations [e.g., 9]. A more recent line assumes access to a known differentiable verifier and uses it to rebalance model density toward valid regions via verifier-constrained entropy maximization [13]. In contrast, we assume access only to black-box, unknown verifiers, covering highly-relevant settings in science, where feedback is experimental and/or non-differentiable. Moreover, ACTFLOW expands valid coverage through directed synthetic data generation rather than verifier-constrained optimization.

**Safe (active) exploration theory** Safe exploration studies learning under unknown safety constraints. Early work uses Gaussian-process confidence bounds to restrict exploration to points certified above a safety threshold [50, 51]. Subsequent safe-RL methods extend this principle to MDPs and control [e.g., 53, 6, 28]. We bridge *safety* in safe exploration with *validity* in generative modeling: expanding a generable set amounts to discovering new valid (i.e., safe) regions. In scientific discovery, invalid samples lead to rejected verifier queries, not unsafe actions in safety-critical systems; thus, ACTFLOW does not perform safe-set constrained planning. Rather, we employ this viewpoint to provide a first-of-its-kind reachability-based theory of out-of-distribution flow modeling.

## 7 Conclusion

We depart from data distribution matching and formulate *out-of-distribution flow modeling* via *generable set expansion*, i.e., expanding the valid region of design space a model samples with non-negligible probability. We introduced ACTFLOW, a continued-pretraining scheme that uses verifier feedback on self-generated data to actively expand in the learned flow representation – leading to first-of-their-kind reachability-based guarantees for out-of-distribution flow modeling. Across small organic molecules, drug-like molecules, therapeutic peptides, and protein sequences, ACTFLOW consistently improves coverage, diversity, and validity over widely adopted recursive self-generation baselines. As for limitations, while our framework allows for task-agnostic expansions toward valid, previously inaccessible regions where new-to-nature discoveries may reside – future work will need to assess whether this form of exploration yields concrete gains (i.e., discoveries) in specific real-world applications.

## Acknowledgments

This publication was made possible by the ETH AI Center doctoral fellowship to Riccardo De Santi. The project has received funding from the Swiss National Science Foundation under NCCR Catalysis (grant number 180544 and 225147) and NCCR Automation (grant agreement 51NF40 180545), National Centres of Competence in Research funded by the Swiss National Science Foundation. This work was supported by an ETH Zurich Research Grant. This research was also supported by a grant from the High-throughput Institute for Discovery (HIT-ID) at the University of Pennsylvania to the laboratory of Pranam Chatterjee.

**Competing Interests.** P.C. is a co-founder of Gameto, Inc., UbiquiTx, Inc., AtomBioworks, Inc., and Kosha Sciences, Inc. and advises companies involved in therapeutics development. P.C.'s interests are reviewed and managed by the University of Pennsylvania in accordance with their conflict-of-interest policies.

## References


[1] Silas Alberti, Kenan Hasanaliyev, Manav Shah, and Stefano Ermon. Data unlearning in diffusion models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 3084–3100, 2025.

[2] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] Sina Alemohammad, Ahmed Imtiaz Humayun, Shruti Agarwal, John Collomosse, and Richard Baraniuk. Self-improving diffusion models with synthetic data. *arXiv preprint arXiv:2408.16333*, 2024.

[4] Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

[5] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

[6] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

[7] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

[8] 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.

[9] Onur Celik, Zechu Li, Denis Blessing, Ge Li, Daniel Palenicek, Jan Peters, Georgia Chalvatzaki, and Gerhard Neumann. DIME: Diffusion-based maximum entropy reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

[10] Yongcan Chen, Ruyun Hu, Keyi Li, Yating Zhang, Lihao Fu, Jianzhi Zhang, and Tong Si. Deep mutational scanning of an oxygen-independent fluorescent protein creilov for comprehensive profiling of mutational and epistatic effects. *ACS Synthetic Biology*, 12(5):1461–1473, 2023.

[11] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[12] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations (ICLR)*, 2023.

[13] Riccardo De Santi, Kimon Protopapas, Ya-Ping Hsieh, and Andreas Krause. Verifier-constrained flow expansion for discovery beyond the data. In *International Conference on Learning Representations (ICLR)*, April 2026.

[14] Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Flow density control: Generative optimization beyond entropy-regularized fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[15] Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Provable maximum entropy manifold exploration via diffusion models. In *International Conference on Machine Learning*, 2025.

[16] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

[17] Ian Dunn and David Ryan Koes. Mixed continuous and categorical flow matching for 3d de novo molecule generation. *ArXiv*, pages arXiv–2404, 2024.

[18] Aaron L. Feller and Claus O. Wilke. Peptide-specific chemical language model successfully predicts membrane diffusion of cyclic peptides. *bioRxiv*, 2024.

[19] Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *Trans. Mach. Learn. Res.*, 2023, 2022.

[20] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

[21] Jeff Guo and Philippe Schwaller. Tango: direct optimization of constrained synthesizability for generative molecular design. *Nature Computational Science*, 2026.

[22] Sven Gutjahr, Riccardo De Santi, Luca Schaufelberger, Kjell Jorner, and Andreas Krause. Constrained molecular generation via sequential flow model fine-tuning. In *International Conference on Machine Learning (ICML)*, 2026.

[23] Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

[24] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

[26] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022.

[27] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[28] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *2018 IEEE conference on decision and control (CDC)*, pages 6059–6066. IEEE, 2018.

[29] Béatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

[30] Xinhao Li and Denis Fourches. Smiles pair encoding: a data-driven substructure tokenization algorithm for deep learning. *Journal of chemical information and modeling*, 61(4):1560–1569, 2021.

[31] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[32] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations (ICLR)*, 2023.

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

[34] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[35] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *International Conference on Learning Representations (ICLR)*, 2024.

[36] Dang Nguyen, Jiping Li, Jinghao Zheng, and Baharan Mirzasoleiman. Do we need all the synthetic data? targeted image augmentation via diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[37] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *International Conference on Learning Representations (ICLR)*, 2025.

[38] Barna Pásztor, Parnian Kassraie, and Andreas Krause. Bandits with preference feedback: A stackelberg game perspective. *Advances in Neural Information Processing Systems*, 37:11997–12034, 2024.

[39] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. *International Conference on Learning Representations (ICLR)*, 2023.

[40] Cristian Perez Jensen, Luca Schaufelberger, Riccardo De Santi, Kjell Jorner, and Andreas Krause. Value matching: Scalable and gradient-free reward-guided flow adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026.

[41] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[42] Jarrid Rector-Brooks, Théophile Lambert, Marta Skreta, Daniel Roth, Yueming Long, Zi-Qi Li, Xi Zhang, Miruna Cretu, Francesca-Zhoufan Li, Tanvi Ganapathy, et al. General multimodal protein design enables dna-encoding of chemistry. *arXiv preprint arXiv:2604.05181*, 2026.

[43] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[44] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[45] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

[46] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[48] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[49] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2021.

[50] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *International conference on machine learning*, pages 997–1005. PMLR, 2015.

[51] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. In *International conference on machine learning*, pages 4781–4789. PMLR, 2018.

[52] Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *42nd International Conference on Machine Learning*, 2025.

[53] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. *Advances in neural information processing systems*, 29, 2016.

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

[55] Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

[56] Lei Wang, Nanxi Wang, Wenping Zhang, Xurui Cheng, Zhibin Yan, Gang Shao, Xi Wang, Rui Wang, and Caiyun Fu. Therapeutic peptides: current applications and future directions. *Signal transduction and targeted therapy*, 7(1):48, 2022.

[57] 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, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[58] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, 1988.

[59] Jason Yang, Wenda Chu, Daniel Khalil, Raul Astudillo, Bruce J. Wittmann, Frances H. Arnold, and Yisong Yue. Steering generative models with experimental data for protein fitness optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[60] Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *Advances in Neural Information Processing Systems*, 37:73366–73398, 2024.

[61] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *International Conference on Learning Representations (ICLR)*, 2025.

# A Appendix

## Contents


**B Central Limitation of Standard Pre-training with an Imperfect Model** **16**

**C A Warm-Up Analysis of Failure Modes of Expansion via Synthetic Data** **16**

C.1 Compact Analysis . . . . . 16

C.2 Extensive Analysis with Proofs . . . . . 17

**D Gradient Descent and Ascent on flow matching loss** **21**

D.1 Signed replay-based continued pre-training and connection to data unlearning . . . 21

**E Intermezzo: Theory of Active Safe Logistic Regression** **24**

**F Theoretical Analysis of Diffusion Models Active Diffusion Expansion** **28**

F.1 Probabilistic Modeling of Binary Verifier over Generative Model Learned Representation . . . . . 28

F.2 Abstraction of the Generative Model as an EBM . . . . . 30

F.3 Active Diffusion Expansion Core Analysis . . . . . 31

F.4 Deriving a Lower Bound on Validity . . . . . 34

F.5 Proof of design-space coverage corollary . . . . . 34

F.6 Full design-space coverage under global reachability . . . . . 35

**G ActFlow for Discrete Diffusion** **37**

G.1 Discrete Diffusion as Continuous-Time Markov Chains . . . . . 37

G.2 Entropy-Regularized Uncertainty Optimization for Discrete Diffusion . . . . . 37

G.3 Uncertainty-Aware Fine-Tuning of Discrete Diffusion . . . . . 37

**H Experimental Details** **39**

H.1 Domain-agnostic evaluation metrics for real-world OOD generative modeling. . . . 39

H.2 Illustrative 2D Experiments . . . . . 39

H.3 Molecular Design: QM9 Experiments . . . . . 40

H.4 Molecular Design: GEOM-Drugs Experiments . . . . . 41

H.5 Therapeutic Peptide Design Experiments . . . . . 41

H.6 Protein Sequence Design Experiments . . . . . 42

H.7 Sensitivity to regularization, representation timestep, and negative feedback . . . . 43

H.8 Structural validity with PoseBusters . . . . . 44

## B Central Limitation of Standard Pre-training with an Imperfect Model

In the main text, Eq. (6) is stated for clarity under the idealization that the pre-trained model generates only valid designs. In practice, this need not hold: the generable set may contain invalid designs, so that

$$\Omega^{\tau}_{\theta^{\mathrm{pre}}} \cap (\mathcal{X} \setminus \Omega^{\star}) \neq \emptyset.$$

This does not change the central limitation. One simply applies the coverage statement to the *valid part* of the generable set,

$$\Omega^{\tau,\mathrm{val}}_{\theta^{\mathrm{pre}}} := \Omega^{\tau}_{\theta^{\mathrm{pre}}} \cap \Omega^{\star} \subseteq \Omega^{\star}.$$

The realistic limitation is therefore

$$\Omega^{\tau,\mathrm{val}}_{\theta^{\mathrm{pre}}} \subset \Omega^{\star}, \qquad \mathrm{Vol}\Big(\Omega^{\tau,\mathrm{val}}_{\theta^{\mathrm{pre}}}\Big) \ll \mathrm{Vol}(\Omega^{\star}),$$

i.e., the pre-trained model covers only a small fraction of the valid design space with non-negligible probability, even if it also assigns mass to invalid regions.

This is precisely the setting addressed by ACTFLOW. The algorithm does not require the initial generable set to be valid; it only requires verifier feedback to identify which generated samples are valid and to adapt the model toward newly verified valid regions. Thus, generable set expansion should be understood as expansion of the valid, verifier-certified portion of the model's coverage. Moreover, when the initial model has partial validity, reallocating mass toward newly verified valid regions can also increase the model's overall validity, as we observe in our experiments, where the pre-trained models are imperfect and ACTFLOW systematically improves not only coverage and diversity, but typically increases substantially validity as well.

**Volume over the valid design space.** Throughout, $\mathrm{Vol}(\cdot)$ denotes volume with respect to a domain-appropriate reference measure on the valid design space $\Omega^{\star}$. Formally, let $\nu^{\star}$ be a reference measure supported on $\Omega^{\star}$: Lebesgue measure when $\Omega^{\star}$ is full-dimensional, intrinsic Hausdorff measure when $\Omega^{\star}$ is a lower-dimensional manifold, and counting measure in discrete design spaces. For any $A \subseteq \mathcal{X}$, we write

$$\mathrm{Vol}(A) := \nu^{\star}(A \cap \Omega^{\star}).$$

Thus, the central limitation in Eq. (6) measures how much of the valid design space is covered by the model's $\tau$-generable set, rather than ambient Lebesgue volume in $\mathcal{X}$. This avoids degeneracies in settings where valid designs lie on lower-dimensional manifolds or discrete spaces.

## C A Warm-Up Analysis of Failure Modes of Expansion via Synthetic Data

### C.1 Compact Analysis

**Warm Up Gaussian Setting.** A widely adopted approach to synthetic data generation is recursive sampling with closed-loop verifier filtering. The purpose of this section is *not* to establish a general impossibility result for this paradigm, but to use a minimal analytical model to isolate possible failure modes relevant to generable-set expansion. To this end, we introduce a simple probabilistic notion of the *generable valid frontier*: valid samples that remain reliably generable under the current model, yet lie away from its dominant modes. Such points form a natural abstraction of expansion-enabling samples. The analysis then shows that standard recursive self-generation schemes may be unlikely to produce them even in a simple low-dimensional setting, thereby shedding light on possible failure modes of data generation schemes and motivating algorithmic desiderata for expansion. We report derivations in Apx. C.2.

**A Gaussian abstraction of pre-training.** We consider a design space $\mathcal{X} = \mathbb{R}^d$. Let $U \subset \mathbb{R}^d$ be a $k$-dimensional linear subspace, let $m := d - k$, and write $\mathbb{R}^d = U \oplus U^{\perp}$. We interpret $U$ as a low-dimensional region well captured by pre-training, and $U^{\perp}$ as orthogonal directions along which expansion beyond the dominant pre-trained modes could occur. For $x \in \mathbb{R}^d$, write $x = x_U + x_{\perp}$ with $x_U = \Pi_U x$ and $x_{\perp} = \Pi_{U^{\perp}} x$. We model the pre-trained generator $p^{\theta_0}$ by the anisotropic Gaussian

$$X \sim p^{\theta_0} =: p_0 = \mathcal{N}\big(0,\, I_U \oplus \sigma^2 I_{U^{\perp}}\big), \qquad 0 < \sigma \ll 1.$$

Thus, the model is spread along $U$, while it is sharply concentrated around $U$ in the orthogonal directions. For a density threshold $\tau > 0$, the corresponding generable set is

$$\Omega^{\tau}_{\theta_0} = \left\{ x \in \mathbb{R}^d : \|x_U\|_2^2 + \sigma^{-2}\|x_\perp\|_2^2 \le r_\tau^2 \right\}, \qquad r_\tau^2 := 2\log\left(\frac{(2\pi)^{-d/2}\sigma^{-m}}{\tau}\right),$$

namely an ellipsoid elongated along $U$ and very thin along $U^\perp$.

**A probabilistic notion of generable valid frontier.** To reason about samples that may support expansion, we isolate a shell inside $\Omega^{\tau}_{\theta_0}$ that is close to its orthogonal boundary. Fix $R_U > 0$ and radii $0 < \rho_- < \rho_+$ such that $R_U < r_\tau$ and $\rho_+ < \sigma\sqrt{r_\tau^2 - R_U^2}$. We denote the *generable frontier* by

$$\mathcal{F}_{\text{frontier}} := \left\{ x \in \mathbb{R}^d : \|x_U\|_2 \le R_U,\ \rho_- \le \|x_\perp\|_2 \le \rho_+ \right\}.$$

By construction, $\mathcal{F}_{\text{frontier}} \subseteq \Omega^{\tau}_{\theta_0}$. Thus, points in $\mathcal{F}_{\text{frontier}}$ remain reliably generable under the current model, since they lie inside its generable set, while already exhibiting a substantial orthogonal deviation from the model dominant modes. Next, fix a unit vector $u_\star \in U^\perp$ and an opening angle $\phi \in (0, \pi/2)$. We define the cone of *valid directions of expansion*

$$C_\phi(u_\star) := \left\{ y \in U^\perp \setminus \{0\} : \left\langle \frac{y}{\|y\|_2}, u_\star \right\rangle \ge \cos\phi \right\}.$$

We call *generable valid frontier* the set

$$\mathcal{V}_{\text{frontier}} := \{ x \in \mathcal{F}_{\text{frontier}} : x_\perp \in C_\phi(u_\star) \}.$$

Thus $\mathcal{V}_{\text{frontier}}$ consists of samples that lie in the model generable frontier, and are aligned with a valid direction of expansion. While $\mathcal{V}_{\text{frontier}}$ does not provide a general first-principles characterization of all useful samples, it formally captures in a minimal way the qualitative tension we want to study: useful self-generated samples should plausibly be $(i)$ *generable* by the current model, yet $(ii)$ *not* lying within its dominant modes, and $(iii)$ aligned with valid directions of expansion. The next result shows that standard self-generation scheme might sample data within this natural class of frontier-valid samples with extremely low probability even in the introduced illustrative Gaussian generator setting.

**Proposition 1** (Standard self-generation is unlikely to find frontier-valid samples)**.** *Let $X \sim p_0$, and let $\mathcal{V}_{\text{frontier}}$ be defined above. For any $\eta \in (0,1)$, define $\rho_\eta := \sigma\sqrt{m + 2\sqrt{m\log(1/\eta)} + 2\log(1/\eta)}$. Assume this choice is feasible for the frontier shell, i.e. $\rho_\eta < \rho_+$. If $\rho_- = \rho_\eta$, then*

$$\Pr\big(X \in \mathcal{V}_{\text{frontier}}\big) \le \eta \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

*Consequently, for $N$ i.i.d. passive samples $X_1, \dots, X_N \sim p_0$,*

$$\Pr\left(\exists i \in [N] : X_i \in \mathcal{V}_{\text{frontier}}\right) \le N\eta \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

**What the proposition reveals.** The proposition isolates two sources of difficulty in recursive self-generation. The $\eta$-term reflects the rarity of reaching the generable frontier at all, namely of producing a sample with substantial orthogonal deviation from the dominant modes. Beyond this, an additional exponential penalty governs the chance of landing in the correct valid cone. Hence, when valid expansion directions are sparse, the sample complexity of passively obtaining a frontier-valid point becomes exponential in the orthogonal dimension $m$.

The algorithmic implication is clear: closed-loop verifier filtering alone is not enough for reliable expansion. Successful expansion requires actively steering generation toward the generable valid frontier.

### C.2 Extensive Analysis with Proofs

**Gaussian warm-up: self-generation requires rare valid frontier events** We present a simple Gaussian model that isolates the two difficulties behind standard self-generation for efficient out-of-distribution discovery: sampling a *large deviation* away from the low-dimensional region captured by pre-training, and doing so along a *valid direction* of expansion.

We work directly on the design space $\mathcal{X} = \mathbb{R}^d$, so in this warm-up there is no separate representation map and $\mathcal{X} = \mathcal{Z}$. Let $U \subset \mathbb{R}^d$ be a $k$-dimensional linear subspace, and write

$$m := d - k, \qquad \mathbb{R}^d = U \oplus U^\perp.$$

For every $x \in \mathbb{R}^d$, denote by

$$x_U := \Pi_U x, \qquad x_\perp := \Pi_{U^\perp} x,$$

so that $x = x_U + x_\perp$. Throughout this subsection we assume $m \geq 2$, since the directional effect of interest only appears when the orthogonal complement has dimension at least two.

**Pre-trained model.** We model a pre-trained generator $p^{\theta_0}$ by the anisotropic Gaussian

$$X \sim p^{\theta_0} =: p_0 = \mathcal{N}\big(0,\, I_U \oplus \sigma^2 I_{U^\perp}\big), \qquad 0 < \sigma \ll 1.$$

Equivalently,

$$X_U \sim \mathcal{N}(0, I_k), \qquad X_\perp \sim \mathcal{N}(0, \sigma^2 I_m),$$

independently. Its density is

$$p_0(x) = (2\pi)^{-d/2}\sigma^{-m} \exp\left(-\frac{1}{2}\left(\|x_U\|_2^2 + \sigma^{-2}\|x_\perp\|_2^2\right)\right).$$

**Generable set.** Fix a level $\epsilon \in (0, (2\pi)^{-d/2}\sigma^{-m})$. In the sense of Definition 1, the $\epsilon$-level generable set of the pre-trained model is

$$\Omega^\epsilon_{\theta_0} := \{x \in \mathbb{R}^d : p_0(x) \geq \epsilon\}.$$

For the Gaussian above this is the ellipsoid

$$\Omega^\epsilon_{\theta_0} = \left\{x \in \mathbb{R}^d : \|x_U\|_2^2 + \sigma^{-2}\|x_\perp\|_2^2 \leq r_\epsilon^2\right\},$$

where

$$r_\epsilon^2 := 2\log\left(\frac{(2\pi)^{-d/2}\sigma^{-m}}{\epsilon}\right).$$

**Valid direction and generable valid frontier.** Fix a unit vector $u_\star \in U^\perp$ and an opening angle $\phi \in (0, \pi/2)$. We define the one-sided cone of valid orthogonal directions by

$$C_\phi(u_\star) := \left\{y \in U^\perp \setminus \{0\} : \left\langle \frac{y}{\|y\|_2}, u_\star \right\rangle \geq \cos\phi\right\}.$$

We also fix a radius $R_U > 0$ and shell radii $0 < \rho_- < \rho_+$ satisfying

$$R_U < r_\epsilon, \qquad \rho_+ < \sigma\sqrt{r_\epsilon^2 - R_U^2}.$$

The corresponding *generable valid frontier* is

$$\mathcal{V}_{\text{frontier}} := \left\{x \in \mathbb{R}^d : \|x_U\|_2 \leq R_U,\ \rho_- \leq \|x_\perp\|_2 \leq \rho_+,\ x_\perp \in C_\phi(u_\star)\right\}.$$

These are rare points that are still inside the current generable set, but already lie on a specific valid orthogonal frontier along which future continued pre-training may expand.

**Lemma C.1** (The generable valid frontier lies inside the current generable set)**.** *Under the above choice of $R_U, \rho_+, \epsilon$, one has*

$$\mathcal{V}_{\text{frontier}} \subseteq \Omega^\epsilon_{\theta_0}.$$

*Proof.* Take any $x \in \mathcal{V}_{\text{frontier}}$. Then $\|x_U\|_2 \leq R_U$ and $\|x_\perp\|_2 \leq \rho_+$, hence

$$\|x_U\|_2^2 + \sigma^{-2}\|x_\perp\|_2^2 \leq R_U^2 + \sigma^{-2}\rho_+^2 < r_\epsilon^2.$$

By the explicit description of $\Omega^\epsilon_{\theta_0}$, this implies $x \in \Omega^\epsilon_{\theta_0}$. □

**A larger valid design space.** We consider a valid design space that extends beyond the current generable set in the same direction $u_\star$. Fix any $R > \sigma r_\epsilon$, and define

$$\mathcal{V}_{\text{out}} := \left\{ x \in \mathbb{R}^d : \|x_U\|_2 \le R_U,\ \sigma\sqrt{r_\epsilon^2 - \|x_U\|_2^2} < \|x_\perp\|_2 \le R,\ x_\perp \in C_\phi(u_\star) \right\}.$$

We then set

$$S_0 := \left\{ x \in \mathbb{R}^d : \|x_U\|_2 \le R_U,\ \|x_\perp\|_2 \le \rho_- \right\}, \qquad \Omega^\star := S_0 \cup \mathcal{V}_{\text{frontier}} \cup \mathcal{V}_{\text{out}}.$$

Thus $\Omega^\star$ contains an already-valid core $S_0$, a rare but still generable frontier $\mathcal{V}_{\text{frontier}}$, and a genuinely out-of-distribution valid region $\mathcal{V}_{\text{out}} \subset (\Omega^\epsilon_{\theta_0})^c$. Indeed, for any $x \in \mathcal{V}_{\text{out}}$,

$$\|x_U\|_2^2 + \sigma^{-2}\|x_\perp\|_2^2 > \|x_U\|_2^2 + \sigma^{-2}\sigma^2\big(r_\epsilon^2 - \|x_U\|_2^2\big) = r_\epsilon^2,$$

so $x \notin \Omega^\epsilon_{\theta_0}$.

The role of the negative result below is to show that, even before trying to reach $\mathcal{V}_{\text{out}}$, passive self-generation is already unlikely to find the expansion-enabling frontier samples $\mathcal{V}_{\text{frontier}}$.

**A convenient large-deviation scale.** Since $X_\perp \sim \mathcal{N}(0, \sigma^2 I_m)$, one has

$$\frac{\|X_\perp\|_2^2}{\sigma^2} \sim \chi^2_m.$$

Therefore, for any $\eta \in (0,1)$, the choice

$$\rho_\eta := \sigma\sqrt{m + 2\sqrt{m\log(1/\eta)} + 2\log(1/\eta)}$$

satisfies

$$\Pr\big(\|X_\perp\|_2 \ge \rho_\eta\big) \le \eta$$

by the Laurent-Massart inequality [29, 7].

**Proposition 1** (Standard self-generation is unlikely to find frontier-valid samples)**.** *Let $X \sim p_0$, and let $\mathcal{V}_{\text{frontier}}$ be defined above. For any $\eta \in (0,1)$, define $\rho_\eta := \sigma\sqrt{m + 2\sqrt{m\log(1/\eta)} + 2\log(1/\eta)}$. Assume this choice is feasible for the frontier shell, i.e. $\rho_\eta < \rho_+$. If $\rho_- = \rho_\eta$, then*

$$\Pr\big(X \in \mathcal{V}_{\text{frontier}}\big) \le \eta \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

*Consequently, for $N$ i.i.d. passive samples $X_1, \dots, X_N \sim p_0$,*

$$\Pr\left(\exists i \in [N] : X_i \in \mathcal{V}_{\text{frontier}}\right) \le N\eta \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

*Proof.* By definition of $\mathcal{V}_{\text{frontier}}$,

$$\{X \in \mathcal{V}_{\text{frontier}}\} \subseteq \{\|X_\perp\|_2 \ge \rho_-\} \cap \left\{ \left\langle \frac{X_\perp}{\|X_\perp\|_2}, u_\star \right\rangle \ge \cos\phi \right\}.$$

Therefore

$$\Pr\big(X \in \mathcal{V}_{\text{frontier}}\big) \le \Pr\left( \|X_\perp\|_2 \ge \rho_-, \left\langle \frac{X_\perp}{\|X_\perp\|_2}, u_\star \right\rangle \ge \cos\phi \right).$$

Now write

$$X_\perp = \sigma G, \qquad G \sim \mathcal{N}(0, I_m).$$

Let

$$R := \|G\|_2, \qquad S := \frac{G}{\|G\|_2} \in \mathbb{S}^{m-1}.$$

For an isotropic Gaussian, $R$ and $S$ are independent, and $S$ is uniform on the unit sphere $\mathbb{S}^{m-1}$. Since $X_\perp = \sigma G$, the previous probability equals

$$\Pr\left(R \ge \frac{\rho_-}{\sigma}, \langle S, u_\star\rangle \ge \cos\phi\right) = \Pr\left(R \ge \frac{\rho_-}{\sigma}\right) \cdot \Pr\left(\langle S, u_\star\rangle \ge \cos\phi\right),$$

which proves

$$\Pr\big(X \in \mathcal{V}_{\text{frontier}}\big) \le \Pr\big(\|X_\perp\|_2 \ge \rho_-\big) \cdot \Pr\left(\left\langle \frac{X_\perp}{\|X_\perp\|_2}, u_\star \right\rangle \ge \cos\phi\right).$$

For the directional term, a standard spherical-cap bound for $S \sim \mathrm{Unif}(\mathbb{S}^{m-1})$ gives, for every $t \in [0,1]$,

$$\Pr\big(\langle S, u_\star\rangle \ge t\big) \le \exp\left(-\frac{m-1}{2}t^2\right).$$

Applying this with $t = \cos\phi$ yields

$$\Pr\left(\left\langle \frac{X_\perp}{\|X_\perp\|_2}, u_\star \right\rangle \ge \cos\phi\right) = \Pr\big(\langle S, u_\star\rangle \ge \cos\phi\big) \le \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

Combining the two bounds proves

$$\Pr\big(X \in \mathcal{V}_{\text{frontier}}\big) \le \Pr\big(\|X_\perp\|_2 \ge \rho_-\big) \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

If $\rho_- = \rho_\eta$, then by the above $\chi^2$-tail bound,

$$\Pr\big(\|X_\perp\|_2 \ge \rho_\eta\big) \le \eta,$$

hence

$$\Pr\big(X \in \mathcal{V}_{\text{frontier}}\big) \le \eta \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

Finally, for $N$ i.i.d. passive samples $X_1, \dots, X_N$, the union bound gives

$$\Pr(\exists i \in [N] : X_i \in \mathcal{V}_{\text{frontier}}) \le \sum_{i=1}^{N} \Pr(X_i \in \mathcal{V}_{\text{frontier}}) = N \Pr(X \in \mathcal{V}_{\text{frontier}}) \le N\eta \exp\left(-\frac{m-1}{2}\cos^2\phi\right).$$

□

# D Gradient Descent and Ascent on flow matching loss

## D.1 Signed replay-based continued pre-training and connection to data unlearning

This appendix formalizes the signed continued-pretraining update used in Sec. 3 and clarifies its connection to data unlearning objectives for diffusion models [1]. The key point is that, after each active-query round, verifier feedback naturally partitions the queried synthetic data into an *accepted* set and a *rejected* set. This induces a retain/forget decomposition analogous to data unlearning, except that here the partition is generated online by the verifier and used to improve out-of-distribution validity coverage.

**Per-round replay buffers.** At round $t$, let

$$\mathcal{D}_t^+ \subseteq \{x_i : y_i = 1,\ i \le t\}, \qquad \mathcal{D}_t^- \subseteq \{x_i : y_i = 0,\ i \le t\}$$

denote the accepted and rejected replay buffers accumulated up to round $t$. Thus the queried synthetic data available for continued pre-training are partitioned into

$$X_t := \mathcal{D}_t^+ \uplus \mathcal{D}_t^-, \qquad A_t := \mathcal{D}_t^-,$$

where $\uplus$ denotes disjoint union of indexed samples. By construction,

$$A_t \subseteq X_t, \qquad X_t \setminus A_t = \mathcal{D}_t^+.$$

This is exactly the retain/forget decomposition considered in data unlearning: $X_t$ is the current empirical dataset, $A_t$ is the subset to be unlearned, and $X_t \setminus A_t$ is the retained subset.

**Underlying flow-matching loss.** Let $\ell(\theta; x)$ denote the standard flow-matching training loss on a sample $x \in \mathcal{X}$, namely the same loss used in the original pre-training objective. For any finite empirical collection $S$, define

$$L_S(\theta) := \frac{1}{|S|} \sum_{x \in S} \ell(\theta; x). \tag{21}$$

In particular,

$$L_{X_t \setminus A_t}(\theta) = L_{\mathcal{D}_t^+}(\theta), \qquad L_{A_t}(\theta) = L_{\mathcal{D}_t^-}(\theta).$$

**Reduction to the deletion objective.** A first natural objective is to train the model as if the rejected samples had been removed from the replay set, namely

$$L_t^{\mathrm{del}}(\theta) := L_{X_t \setminus A_t}(\theta) = L_{\mathcal{D}_t^+}(\theta). \tag{22}$$

Writing

$$m_t := |\mathcal{D}_t^+|, \qquad k_t := |\mathcal{D}_t^-|, \qquad n_t := m_t + k_t,$$

the same algebra used in [1] gives the exact decomposition

$$L_t^{\mathrm{del}}(\theta) = \frac{n_t}{m_t} L_{X_t}(\theta) - \frac{k_t}{m_t} L_{A_t}(\theta) = \frac{n_t}{m_t} L_{X_t}(\theta) - \frac{k_t}{m_t} L_{\mathcal{D}_t^-}(\theta). \tag{23}$$

Thus, even the valid-only objective can be rewritten using *both* accepted and rejected samples. This is precisely the same deletion identity underlying SISS in [1], specialized here to the verifier-induced partition of queried synthetic data. In the terminology of [1], our practical implementation uses the corresponding *non-importance-sampled* variant, i.e., the analogue of SISS (No IS), rather than the one-pass importance-sampled estimator.

**Why we use a signed objective.** For out-of-distribution generable-set expansion, Eq. (22) is often too conservative: it treats rejected samples only as points to be discarded. In contrast, rejected verifier queries contain useful information about directions in design space toward which the current model should allocate *less* probability mass. This motivates the signed replay objective used in the main text:

$$L_t^{\mathrm{signed}}(\theta) := L_{\mathcal{D}_t^+}(\theta) - \alpha_t\, L_{\mathcal{D}_t^-}(\theta). \tag{24}$$

Its gradient is exactly the signed update:

$$\nabla_\theta L_t^{\mathrm{signed}}(\theta_t) = \nabla_\theta L_{\mathcal{D}_t^+}(\theta_t) - \alpha_t \nabla_\theta L_{\mathcal{D}_t^-}(\theta_t). \tag{25}$$

At the minibatch level, our implementation follows the *non-importance-sampled* variant: we draw separate minibatches

$$U_t^+ \subseteq \mathcal{D}_t^+, \qquad U_t^- \subseteq \mathcal{D}_t^-,$$

and form the empirical losses

$$\widehat{L}_t^+(\theta) = \frac{1}{|U_t^+|} \sum_{x \in U_t^+} \ell(\theta; x), \qquad \widehat{L}_t^-(\theta) = \frac{1}{|U_t^-|} \sum_{x \in U_t^-} \ell(\theta; x).$$

The practical signed update is then

$$g_t = \nabla \widehat{L}_t^+(\theta_t) - \alpha_t \nabla \widehat{L}_t^-(\theta_t).$$

Hence accepted samples provide an attractive update, while rejected samples provide a repulsive update of controlled magnitude. This corresponds to the analogue of SISS (No IS) in [1]: two separate replay-buffer minibatches are used, rather than a single importance-sampled mixture minibatch.

**Relation to weighted data-unlearning objectives.** The weighted SISS objective of [1] can be written as

$$L^{\mathrm{wSISS}}(\theta) = L_{X \setminus A}(\theta) - s \frac{|A|}{|X \setminus A|} L_A(\theta). \tag{26}$$

Applying this to the verifier-induced partition

$$X = X_t = \mathcal{D}_t^+ \uplus \mathcal{D}_t^-, \qquad A = A_t = \mathcal{D}_t^-,$$

yields

$$L_t^{\mathrm{wSISS}}(\theta) = L_{\mathcal{D}_t^+}(\theta) - s_t \frac{k_t}{m_t} L_{\mathcal{D}_t^-}(\theta). \tag{27}$$

Therefore our signed replay objective in Eq. (24) is exactly the same class of weighted retain-minus-forget objective, with the identification

$$\alpha_t = s_t \frac{k_t}{m_t}. \tag{28}$$

In this sense, the update used by ACTFLOW is a direct reduction of the weighted unlearning objective of [1] to the online verifier-guided expansion setting considered here.

**Possible importance sampling (IS) extension.** As mentioned, our implementation is the direct analogue of SISS (No IS) in [1], rather than the one-pass importance-sampled estimator introduced there. An importance-sampled extension could also be used in our setting by sampling from a suitable mixture over accepted and rejected replay samples and reweighting by the corresponding importance ratios, but we avoid this additional estimator complexity here for simplicity.

**Interpretation in our setting.** The reduction above is exact at the level of the empirical replay buffers held fixed at round $t$. Conditionally on the current buffers $(\mathcal{D}_t^+, \mathcal{D}_t^-)$, the signed objective

$$L_t^{\mathrm{signed}}(\theta) = L_{\mathcal{D}_t^+}(\theta) - \alpha_t L_{\mathcal{D}_t^-}(\theta)$$

should be read as follows:

1. the first term increases likelihood of synthetic samples that have been certified as valid by the verifier and therefore support expansion into new valid regions;
2. the second term decreases likelihood of queried samples that were explicitly rejected by the verifier and therefore encode evidence against those directions of expansion.

Thus, unlike standard valid-only pre-training, the signed replay update uses *both* outcomes of the verifier query and converts the online generation-and-verification loop into a principled retain/forget training signal.

**Practical choice of $\alpha_t$.** In the practical implementation, the loss-level weight $\alpha_t$ is calibrated online as a gradient-norm fraction. Concretely, for a target ratio $\alpha \in [0, 1]$, we set

$$\alpha_t \;:=\; \alpha \frac{\|\nabla \widehat{L}_t^+(\theta_t)\|_2}{\|\nabla \widehat{L}_t^-(\theta_t)\|_2},$$

This yields

$$\left\|\alpha_t \nabla \widehat{L}_t^-(\theta_t)\right\|_2 \approx \alpha \left\|\nabla \widehat{L}_t^+(\theta_t)\right\|_2,$$

so the rejected-sample contribution is scaled to have a prescribed norm fraction relative to the accepted-sample contribution.

## E Intermezzo: Theory of Active Safe Logistic Regression

Suppose that given any query point $x$, the verifier provides binary feedback $y \in \{0, 1\}$ generated according to

$$\Pr[y = 1 \mid x] = s(f(x)), \tag{29}$$

where $s : \mathbb{R} \to [0, 1]$ is the sigmoid function, and $f : \mathcal{X} \to \mathbb{R}$ is a validity function. We assume that $f$ lies in the RKHS $\mathcal{H}_k$ associated with a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with bounded norm $\|f\|_k \le B$. We also assume that $f$ is $L_f$-smooth with respect to a metric $d$, namely

$$|f(x) - f(x')| \le L_f d(x, x') \qquad \forall x, x' \in \mathcal{X}.$$

We adopt a *pre-query* indexing convention. At the beginning of round $t$, the learner has access to the history

$$H_{t-1} \triangleq \{(x_\tau, y_\tau)\}_{\tau=1}^{t-1},$$

fits a probabilistic model using $H_{t-1}$, constructs a safe set $S_t$, and then selects the next query $x_t$. Given $H_{t-1}$, the learner estimates $f$ by minimizing the regularized negative log-likelihood

$$\begin{aligned} \mu_t &\triangleq \operatorname*{arg\,min}_{g \in \mathcal{H}_k,\ \|g\|_k \le B} \mathcal{L}(g, H_{t-1}), \\ \mathcal{L}(g, H_{t-1}) &\triangleq \sum_{\tau=1}^{t-1} \Big[ -y_\tau \log\big(s(g(x_\tau))\big) - (1 - y_\tau) \log\big(1 - s(g(x_\tau))\big) \Big] + \frac{\lambda}{2} \|g\|_k^2 , \end{aligned} \tag{30}$$

where $\lambda > 0$ is a regularization coefficient. By the Representer Theorem [44], the solution lies in the span of the kernel sections at the previously queried points.

Given $\mu_t$, the learner predicts the verifier output via $s(\mu_t(x))$. Previous work gives anytime-valid confidence sets of the form

$$[s(\mu_t(x)) \pm \beta_t(\delta)\sigma_t(x)]$$

for kernelized logistic regression [38]. Under our indexing convention, the uncertainty scores are

$$\sigma_t^2(x) = k(x, x) - k_{t-1}^\top(x) \left(K_{t-1} + \lambda\kappa I_{t-1}\right)^{-1} k_{t-1}(x),$$

where

$$\kappa \triangleq \sup_{a \le B} \frac{1}{\dot{s}(a)},$$

$k_{t-1}(x)$ is the kernel vector with entries $[k_{t-1}(x)]_i = k(x_i, x)$, and $K_{t-1}$ is the kernel matrix with entries $[K_{t-1}]_{i,j} = k(x_i, x_j)$.

We will use the following pre-query reindexing of the confidence-sequence guarantee.

**Theorem E.1** (Kernelized Logistic Confidence Sequences)**.** *Assume $f \in \mathcal{H}_k$ and $\|f\|_k \le B$. Assume that the data $x_1, \dots, x_t$ used to fit the model $\mu_t$ lie in a compact subset $A \subseteq \mathcal{X}$. Assume further that $k(x, x') \le 1$ for $x, x' \in A$. Let $0 < \delta < 1$ and define*

$$\beta_t(\delta) \triangleq 4L_s B + 2L_s \sqrt{\frac{2\kappa}{\lambda}\Big(\gamma_{t-1}^A + \log(1/\delta)\Big)}, \tag{31}$$

*where*

$$\gamma_t^A \triangleq \max_{x_1, \dots, x_t \in A} \frac{1}{2} \log\det\Big(I_t + (\lambda\kappa)^{-1} K_t\Big), \qquad L_s \triangleq \sup_{a \le B} \dot{s}(a).$$

*Then*

$$\Pr\big(\forall t \ge 1,\ \forall x \in A :\ \big|s(\mu_t(x)) - s(f(x))\big| \le \beta_t(\delta)\, \sigma_t(x)\big) \ge 1 - \delta.$$

Using these confidence sets, the learner may perform active safe logistic regression, querying only points that are certified to satisfy $s(f(x)) \ge h$ for some threshold $h \in [0, 1]$. This parallels safe Bayesian optimization in the regression setting [e.g., 50, 51].

In this setting, safe designs are those for which the verifier returns label 1 with sufficiently high probability. Suppose the learner is given a safe seed set $S_0 \subseteq \mathcal{X}$. At round $t$, the learner updates its certified safe set as

$$S_t = S_{t-1} \cup \{x \in \mathcal{X} \mid \exists x' \in S_{t-1} : s(\mu_t(x')) - \beta_t(\delta)\sigma_t(x') - L_s L_f d(x, x') \ge h\}. \tag{32}$$

The next lemma shows that, conditioned on the confidence event of Theorem E.1, every point ever added to $S_t$ is indeed safe.

**Lemma E.2** (Safety)**.** *Condition on the event that the confidence intervals of Theorem E.1 are valid. Then for every $t \geq 0$, every $x \in S_t$ satisfies $s(f(x)) \geq h$.*

*Proof.* We argue by induction on $t$. The claim holds at $t = 0$ by assumption on the seed set $S_0$. Now fix $t \geq 1$, and let $x \in S_t$. Either $x \in S_{t-1}$, in which case the claim follows by induction, or

$$x \in \{x \in \mathcal{X} \mid \exists x' \in S_{t-1} : s(\mu_t(x')) - \beta_t(\delta)\sigma_t(x') - L_s L_f d(x, x') \geq h\}.$$

In the latter case, there exists $x' \in S_{t-1}$ such that

$$s(\mu_t(x')) - \beta_t(\delta)\sigma_t(x') - L_s L_f d(x, x') \geq h.$$

By the confidence event,

$$s(f(x')) \geq s(\mu_t(x')) - \beta_t(\delta)\sigma_t(x').$$

Moreover, since $s$ is $L_s$-Lipschitz and $f$ is $L_f$-Lipschitz,

$$s(f(x)) \geq s(f(x')) - L_s L_f d(x, x').$$

Combining the two inequalities gives

$$s(f(x)) \geq s(\mu_t(x')) - \beta_t(\delta)\sigma_t(x') - L_s L_f d(x, x') \geq h.$$

Thus every $x \in S_t$ is safe. □

Notice that by construction, the sequence $\{S_t\}_{t \geq 0}$ is monotone:

$$S_{t-1} \subseteq S_t \qquad \forall t \geq 1.$$

Using this set of safe decisions, the learner may follow an active learning procedure which samples the next point at which to query the verifier as the one in the set of safe decisions that has the highest uncertainty score, often referred to as *safe uncertainty sampling*:

$$x_t = \arg\max_{x \in S_t} \sigma_t(x). \tag{33}$$

After querying the verifier at $x_t$ and observing $y_t$, the history updates as

$$H_t = H_{t-1} \cup \{(x_t, y_t)\}.$$

We ultimately want to show some notion of suitable expansion of the safe set by following the above sampling procedure. As our safety set at any time is defined as an expansion of the safe set at the previous time, we cannot guarantee convergence to the entire subset of $\mathcal{X}$ which is safe. Instead, we must define some notion of *reachable safe set* of decisions. To this end, we consider the tightened one step reachability operator defined as

$$R_\varepsilon(S) \triangleq \{x \in \mathcal{X} \mid \exists x' \in S : s(f(x')) - L_s L_f d(x, x') - \varepsilon \geq h\}.$$

The $H$-fold recursive application of this operator is denoted by $R_\varepsilon^H(S)$, and its closure as $H \to \infty$ is denoted $\bar{R}_\varepsilon(S)$. Our goal is to show that safe uncertainty sampling expands the certified safe set toward $\bar{R}_0(S_0)$. We first prove an uncertainty-reduction bound over the current safe set. Let

$$\Omega^\star \triangleq \{x \in \mathcal{X} : s(f(x)) \geq h\}$$

denote the subset of $\mathcal{X}$ corresponding to the *valid design space*, and assume that $\Omega^\star$ is compact. Without loss of generality, by normalization of the kernel, assume that $k(x, x') \leq 1$ for $x, x' \in \Omega^\star$.

**Lemma E.3** (Uncertainty Reduction)**.** *Fix $t \geq 0$ and $T \geq 1$. Under safe uncertainty sampling (33), the epistemic uncertainty at time $t + T$ decays as*

$$\max_{x \in S_t} \sigma_{t+T}(x) \leq \sqrt{\frac{2}{\log(1 + (\lambda\kappa)^{-1})}} \sqrt{\frac{\gamma_T^{\Omega^\star}}{T}}.$$

*Proof.* By monotonicity of the uncertainty score, it holds that

$$\begin{aligned}
\max_{x\in S_t}\sigma_{t+T}(x) &\le \frac{1}{T}\sum_{n=0}^{T-1}\max_{x\in S_t}\sigma_{t+n}(x) && \text{(uncertainty is monotone)}\\
&\le \frac{1}{T}\sum_{n=0}^{T-1}\max_{x\in S_{t+n}}\sigma_{t+n}(x) && (S_t \text{ are monotone})\\
&= \frac{1}{T}\sum_{n=0}^{T-1}\sigma_{t+n}(x_{t+n}) && \text{(safe uncertainty sampling (33))}\\
&\le \frac{1}{\sqrt{T}}\sqrt{\sum_{n=0}^{T-1}\sigma_{t+n}^2(x_{t+n})} && \text{(Cauchy–Schwarz).}
\end{aligned}$$

It holds that $x \le \frac{\bar{x}}{\log(1+\bar{x})}\log(1+x)$ for $0 \le x \le \bar{x}$. Let $K_{t:t+T-1}$ be the Gram matrix of the queried points $x_t, \dots, x_{t+T-1}$, let $K_{1:t-1}$ be the Gram matrix of the previously queried points $x_1, \dots, x_{t-1}$, let $K_{t:t+T-1,\,1:t-1}$ be the corresponding cross-Gram matrix, and define

$$\Sigma_{t:t+T-1|1:t-1} \triangleq K_{t:t+T-1} - K_{t:t+T-1,\,1:t-1}(K_{1:t-1}+\lambda\kappa I_{t-1})^{-1}K_{1:t-1,\,t:t+T-1}.$$

Then the above inequality along with the bound $k(x,x') \le 1$ for $x, x' \in \Omega^\star$ implies that

$$\begin{aligned}
\max_{x\in S_t}\sigma_{t+T}(x) &\le \frac{1}{\sqrt{T}}\sqrt{\frac{1}{\log(1+(\lambda\kappa)^{-1})}}\sqrt{\sum_{n=0}^{T-1}\log\left(1+\frac{\sigma_{t+n}^2(x_{t+n})}{\lambda\kappa}\right)}\\
&= \frac{1}{\sqrt{T}}\sqrt{\frac{1}{\log(1+(\lambda\kappa)^{-1})}}\sqrt{\log\det(I_T+(\lambda\kappa)^{-1}\Sigma_{t:t+T-1|1:t-1})}\\
&\le \frac{1}{\sqrt{T}}\sqrt{\frac{1}{\log(1+(\lambda\kappa)^{-1})}}\sqrt{\log\det(I_T+(\lambda\kappa)^{-1}K_{t:t+T-1})}\\
&\le \frac{1}{\sqrt{T}}\sqrt{\frac{1}{\log(1+(\lambda\kappa)^{-1})}}\sqrt{\max_{x_1,\dots,x_T\in\Omega^\star}\log\det(I_T+(\lambda\kappa)^{-1}K_T)}\\
&= \sqrt{2}\frac{\sqrt{\gamma_T^{\Omega^\star}}}{\sqrt{T}}\sqrt{\frac{1}{\log(1+(\lambda\kappa)^{-1})}}.
\end{aligned}$$

□

The next lemma shows that once the uncertainty over the current safe set is sufficiently small, additional safe uncertainty sampling expands the safe set according to the one-step reachability operator.

**Lemma E.4** (One-step Reachable Expansion)**.** *Consider $\varepsilon > 0$, $t \ge 0$, and $T \ge 1$. Suppose that*

$$T \ge \frac{8\beta_{t+T}^2\gamma_T^{\Omega^\star}\log(1+(\lambda\kappa)^{-1})}{\varepsilon^2}.$$

*Then, conditioned on the event that the intervals of Theorem E.1 are valid, it holds that*

$$R_\varepsilon(S_t) \subseteq S_{t+T}.$$

*Proof.* By Lemma E.3 and the condition on $T$,

$$\max_{x\in S_t}\sigma_{t+T}(x) \le \frac{\varepsilon}{2\beta_{t+T}}. \tag{34}$$

Consider any point $x \in R_\varepsilon(S_t)$. By definition, there exists $x' \in S_t$ such that

$$h \le s(f(x')) - \varepsilon - L_s L_f d(x,x').$$

By the confidence event at round $t+T$,

$$s(f(x')) \leq s(\mu_{t+T}(x')) + \beta_{t+T}\sigma_{t+T}(x').$$

Hence

$$\begin{aligned} h &\leq s(f(x')) - \varepsilon - L_s L_f d(x, x') \\ &\leq s(\mu_{t+T}(x')) + \beta_{t+T}\sigma_{t+T}(x') - \varepsilon - L_s L_f d(x, x') \\ &\leq s(\mu_{t+T}(x')) - \beta_{t+T}\sigma_{t+T}(x') - L_s L_f d(x, x'), \end{aligned}$$

where the last step uses (34), since $x' \in S_t$ implies

$$\beta_{t+T}\sigma_{t+T}(x') \leq \frac{\varepsilon}{2}.$$

By monotonicity of the safe sets, $x' \in S_t \subseteq S_{t+T-1}$. Therefore there exists $x' \in S_{t+T-1}$ such that

$$s(\mu_{t+T}(x')) - \beta_{t+T}\sigma_{t+T}(x') - L_s L_f d(x, x') \geq h.$$

By the definition of $S_{t+T}$ in (32), this implies $x \in S_{t+T}$. □

Applying the above result recursively leads to $H$-step expansion of $S_0$ at time $T^\star$. In particular, suppose $T^\star = HT$ for some $T$ satisfying

$$T \geq \frac{8\beta_{t+T}^2 \gamma_T^{\Omega^\star} \log(1 + (\lambda\kappa)^{-1})}{\varepsilon^2}.$$

This implies that the condition of the above lemma holds for each interval of length $T$, since $\beta_t$ is monotonically increasing. Consequently, it holds that

$$R_\varepsilon^H(S_0) \subseteq S_{T^\star}.$$

We can find $T^\star$ large enough to satisfy this condition as long as the complexity term $\beta_t^2 \gamma_t^{\Omega^\star}$ grows sublinearly with $t$.

# F Theoretical Analysis of Diffusion Models Active Diffusion Expansion

In the following, we first $(i)$ report the general analysis for safe logistic regression that we introduced and presented in Apx. E, where we now re-interpret safety as validity, then we $(ii)$ introduce a probabilistic modeling framework of the generative density $p_t$ via energy-based models, and $(iii)$ derive coverage guarantees with sample complexity for the proposed active expansion algorithm.

## F.1 Probabilistic Modeling of Binary Verifier over Generative Model Learned Representation

We denote by $\phi : \mathcal{X} \to \mathcal{Z}$ the representation map learned by the pre-trained generative model. We now describe the active learning process of the verifier over the learned representation space $\mathcal{Z}$. Suppose that given any query point $x$ with latent representation $z \coloneqq \phi(x)$, the verifier provides binary feedback $y \in \{0, 1\}$ generated according to

$$\Pr[y = 1 \mid z] = s(g(z)), \tag{35}$$

where $s : \mathbb{R} \to [0, 1]$ is the sigmoid function, and $g : \mathcal{Z} \to \mathbb{R}$ is an unknown validity function. We assume that $g$ lies in the RKHS $\mathcal{H}_k$ associated with a kernel $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, with bounded norm $\|g\|_k \leq B$ and that $g$ is $L_g$-smooth with respect to a metric $d$, namely

$$|g(z) - g(z')| \leq L_g d(z, z') \qquad \forall z, z' \in \mathcal{Z},$$

for $B$ and $L_g$ known. We additionally assume there exists a known constant $\bar{Z}$ such that $\int_{\mathcal{Z}} \exp(g(z))dz \leq \bar{Z}$.

On unbounded domains with respect to Lebesgue measure, the energy condition is incompatible with globally bounded kernels such as the squared-exponential kernel. Indeed, if $k$ is bounded on the diagonal and $\|g\|_k \leq B$, then

$$|g(z)| \leq \|g\|_k \sqrt{k(z, z)} \text{ for all } z \in \mathcal{Z}$$

implies that $g$ is uniformly bounded. Hence $\exp(g)$ is bounded below by a positive constant, and therefore cannot be integrable over an infinite-measure domain. Thus, in this setting, the energy condition should be understood as requiring a kernel class capable of representing functions with sufficiently negative tails. For example, sums of bounded kernels with even-degree polynomial kernels yield RKHS that contain coercive negative polynomials, such as $g(z) = c - z^\top Az$ with $A \succ 0$, which satisfy the energy condition.

We adopt a *pre-query* indexing convention. At the beginning of round $t$, the learner has access to the history

$$H_{t-1} \triangleq \{(z_\tau, y_\tau)\}_{\tau=1}^{t-1}.$$

Given $H_{t-1}$, the learner estimates $g$ by minimizing the regularized negative log-likelihood

$$\begin{aligned} \mu_t &\triangleq \operatorname*{arg\,min}_{u \in \mathcal{H}_k,\ \|u\|_k \leq B} \mathcal{L}(u, H_{t-1}), \\ \mathcal{L}(u, H_{t-1}) &\triangleq \sum_{\tau=1}^{t-1} \Big[ -y_\tau \log\big(s(u(z_\tau))\big) - (1 - y_\tau) \log\big(1 - s(u(z_\tau))\big) \Big] + \frac{\lambda}{2} \|u\|_k^2 , \end{aligned} \tag{36}$$

where $\lambda > 0$ is a regularization coefficient. By the Representer Theorem [44], the solution lies in the span of the kernel sections at the previously queried points.

Given $\mu_t$, the learner predicts the verifier output via $s(\mu_t(z))$. Previous work gives anytime-valid confidence sets of the form

$$[s(\mu_t(z)) \pm \beta_t \sigma_t(z)]$$

for kernelized logistic regression [38]. Under our indexing convention, the uncertainty scores are

$$\sigma_t^2(z) = k(z, z) - k_{t-1}^\top(z) \left(K_{t-1} + \lambda\kappa I_{t-1}\right)^{-1} k_{t-1}(z),$$

where

$$\kappa \triangleq \sup_{a \leq B} \frac{1}{\dot{s}(a)},$$

$k_{t-1}(z)$ is the kernel vector with entries $[k_{t-1}(z)]_i = k(z_i, z)$, and $K_{t-1}$ is the kernel matrix with entries $[K_{t-1}]_{i,j} = k(z_i, z_j)$.

We will use the following pre-query reindexing of the confidence-sequence guarantee.

**Theorem F.1** (Kernelized Logistic Confidence Sequences [38]). *Assume $g \in \mathcal{H}_k$ and $\|g\|_k \le B$. Assume that the queried points lie in a compact subset $A \subseteq \mathcal{Z}$. Without loss of generalization, suppose that $k(z, z') \le 1\, \forall z, z' \in A$. Let $0 < \delta < 1$ and define*

$$\beta_t^A(\delta) \triangleq 4L_s B + 2L_s \sqrt{\frac{2\kappa}{\lambda}\Big(\gamma_{t-1}^A + \log(1/\delta)\Big)}, \tag{37}$$

*where*

$$\gamma_t^A \triangleq \max_{z_1,\dots,z_t \in A} \frac{1}{2}\log\det\Big(I_t + (\lambda\kappa)^{-1}K_t\Big), \qquad L_s \triangleq \sup_{a \le B} \dot{s}(a).$$

*Then*

$$\Pr\left(\forall t \ge 1,\ \forall z \in A :\ \big|s(\mu_t(z)) - s(g(z))\big| \le \beta_t^A(\delta)\,\sigma_t(z)\right) \ge 1 - \delta.$$

Using these confidence sets, the learner may perform guarded logistic regression, querying only points that are certified to satisfy $s(g(z)) \ge h$ for some threshold $h \in [0, 1]$. This parallels safe Bayesian optimization in the regression setting [e.g., 50, 51].

In this setting, safe designs are those for which the probabilistic verifier returns label 1 with sufficiently high probability. Suppose the learner is given a safe seed set $S_0 \subseteq \mathcal{Z}$. Given a monotonically increasing sequence of calibration coefficients at level $\delta$, $\left\{\hat{\beta}_t(\delta)\right\}_{t\ge0}$, the set of points that the learner can verify as safe may be expanded at round $t$ as

$$S_t = S_{t-1} \cup \left\{z \in \mathcal{Z} \;\middle|\; \exists z' \in S_{t-1} : s(\mu_t(z')) - \hat{\beta}_t(\delta)\sigma_t(z') - L_s L_g d(z, z') \ge h\right\}. \tag{38}$$

We differentiate between the sequence $\left\{\hat{\beta}_t(\delta)\right\}$ and $\left\{\beta_t^A(\delta)\right\}$ as computing the latter requires knowledge of the subset $A$, whereas our downstream analysis shows that our sampling remains restricted to an unknown set. Despite this, the learner may reasonably have an upper bound on the information capacity of the kernel restricted to the unknown subset, and thus may reasonably have access to such upper bounds.

The next lemma shows that, conditioned on the confidence event of Theorem F.1, every point ever added to $S_t$ satisfies the validity condition.

**Lemma F.2** (Safety). *Condition on the event that the confidence intervals of Theorem F.1 are valid at level $\delta \in (0, 1)$ under a compact set $A$ with $k(z, z') \le 1\, \forall z, z' \in A$ and suppose that the sequence $\left\{\hat{\beta}_t(\delta)\right\}_{t\ge0}$ satisfies $\hat{\beta}_t(\delta) \ge \beta_t^A(\delta)$. Then for every $t \ge 0$, every $z \in S_t$ satisfies $s(g(z)) \ge h$.*

*Proof.* We argue by induction on $t$. The claim holds at $t = 0$ by assumption on the seed set $S_0$.

Now fix $t \ge 1$, and let $z \in S_t$. Either $z \in S_{t-1}$, in which case the claim follows by induction, or

$$z \in \left\{z \in \mathcal{Z} \;\middle|\; \exists z' \in S_{t-1} : s(\mu_t(z')) - \hat{\beta}_t(\delta)\sigma_t(z') - L_s L_g d(z, z') \ge h\right\}.$$

In the latter case, there exists $z' \in S_{t-1}$ such that

$$s(\mu_t(z')) - \hat{\beta}_t(\delta)\sigma_t(z') - L_s L_g d(z, z') \ge h.$$

By the confidence event,

$$s(g(z')) \ge s(\mu_t(z')) - \hat{\beta}_t(\delta)\sigma_t(z').$$

Moreover, since $s$ is $L_s$-Lipschitz and $g$ is $L_g$-Lipschitz,

$$s(g(z)) \ge s(g(z')) - L_s L_g d(z, z').$$

Combining the two inequalities gives

$$s(g(z)) \ge s(\mu_t(z')) - \hat{\beta}_t(\delta)\sigma_t(z') - L_s L_g d(z, z') \ge h.$$

Thus every $z \in S_t$ exceeds the validity threshold. □

Notice that by construction, the sequence $\{S_t\}_{t\ge0}$ is monotone:

$$S_{t-1} \subseteq S_t \qquad \forall t \ge 1.$$

**Scope of the feedback model.** The logistic feedback model is an analytical abstraction of stochastic validity feedback, as may arise in experimental scientific discovery. Our theoretical guarantees are conditional on this feedback model and the stated regularity assumptions. The experiments instead use deterministic proxy verifiers, and therefore assess the practical behavior of ACTFLOW beyond the exact feedback assumptions of the analysis.

### F.2 Abstraction of the Generative Model as an EBM

Our primary abstraction of the generative model is that it approximately tracks the sequence of certified safe sets constructed with a sequence of calibration parameters $\left\{\hat{\beta}_t(\delta)\right\}$. We achieve this abstraction using an energy based model. Specifically, for a fixed $\gamma > 0$ and $0 < \ell \leq h$, we consider an energy function which satisfies

$$\mu_t' \in \Bigg\{ f : \int_{\mathcal{Z}} e^{f(z)} dz \leq \bar{Z}, \\ f(z) \geq \operatorname{logit}(h) \mathbf{1}(S_t) \, \forall z \in \mathcal{Z}, \\ f(z) \leq \operatorname{logit}(\ell) \text{ for all } z \text{ such that } d(z, S_t) \geq \gamma \Bigg\}. \tag{39}$$

The lower bound on $f$ captures the behavior where the updates to our generative model maintain high density on regions that our verifier certifies as valid according to (32). The upper bound on $f$ instead captures the behavior where the model maintains low density on points far from this validated set. The explicit bound on the partition function $\bar{Z}$ ensures that the normalization constant does not cause the density placed on the valid region to vanish. These capture the desiderata that our generative model updates maintain large density in regions which have been verified as valid, and maintain small density in regions far away from this verified region. The generative model updates using both positive and negative samples, defined in Apx. D.1, are designed to satisfy these desiderata. For our theoretical analysis, we assume that this abstraction is valid.

**Assumption F.1.** *Consider $\gamma > 0$ and let $\ell$ be such that*

$$\frac{\exp(\operatorname{logit}(\ell))}{\underline{Z}} \leq \frac{\exp(\operatorname{logit}(h))}{\bar{Z}}. \tag{40}$$

*where $\underline{Z} := \mathsf{vol}(S_0) \exp(\operatorname{logit}(h))$ and $\operatorname{logit}(u) := \log\left(\frac{u}{1-u}\right)$. We assume that the generative model can be abstracted as an EBM by considering the density*

$$p_t(z) = \frac{\exp(\mu_t'(z))}{Z_t}, \qquad Z_t := \int_{z \in \mathcal{Z}} \exp(\mu_t'(z)) \, \mathrm{d}z. \tag{41}$$

*for the energy function $\mu_t'$ defined in (39).*

We define the corresponding *generable set* at level $\tau$ as the high-probability set

$$\Omega_t^\tau := \{x \in \mathcal{X} : p_t(z) \geq \tau\}. \tag{42}$$

The EBM-abstraction of the generative model allows us to draw a relationship between the generable set of the EBM and the certified safe sets in which the generable set at some level $\tau$ and round $t$ contains the set $S_t$.

**Lemma F.3** (Generable and Verifier Set Inequality for the Recursive Safe Set)**.** *Suppose that Assumption F.1 holds with some $\gamma > 0$.*

*Then, for any*

$$\tau \leq \frac{\exp(\operatorname{logit}(h))}{\bar{Z}}, \tag{43}$$

*$S_t$ is a subset of the high-probability set $\Omega_t^\tau$ for the EBM model defined by Assumption F.1, namely $S_t \subseteq \Omega_t^\tau$.*

*Proof.* By definition of the high-probability set,

$$\Omega_t^\tau = \{x \in \mathcal{X} : p_t^\pi(x) \geq \tau\} = \left\{x \in \mathcal{X} : \frac{\exp(\mu_t'(x))}{Z_t} \geq \tau\right\} = \{x \in \mathcal{X} : \mu_t'(x) \geq \log\tau + \log Z_t\}. \tag{44}$$

Now let $z \in S_t$. By definition of $\mu_t'$, $\mu_t'(z)) \geq \text{logit}(h)$. Therefore, a sufficient condition for $x \in \Omega_t^\tau$ is $\text{logit}(h) \geq \log\tau + \log Z_t$, or equivalently, $\tau \leq \exp(\text{logit}(h))/Z_t$.

It remains to upper bound $Z_t$. This holds immediately by definition of $\mu_t'$ such that it satisfies

$$Z_t = \int_{\mathcal{Z}} \exp(\mu_t'(z))dz \leq \bar{Z}.$$

Plugging this bound in above, we find that a sufficient condition is,

$$\tau \leq \frac{\exp(\text{logit}(h))}{\bar{Z}}. \tag{45}$$

Under this condition, every $z \in S_t$ satisfies $\mu_t(z) \geq \log\tau + \log Z_t$, hence by (44) we have $z \in \Omega_t^\tau$. Therefore, $S_t \subseteq \Omega_t^\tau$, concluding the proof. □

At the same time, we can show that the generable set lies within an extension of the valid design space. To this end, we define the inflation of the valid design space.

**Definition 3.** *We define the inflation of the valid set by amount $\zeta \in \mathbb{R}$ in metric $d$ as*

$$\Gamma(\zeta) = \{z \in \mathcal{Z} \setminus \Omega_* : d(z, \Omega_*) \leq \zeta\}.$$

**Assumption F.2.** *Let $\gamma > 0$ be the same as Assumption F.1. We suppose that the set $\Omega_* \cup \Gamma(\gamma)$ is compact. Further assume without loss of generality (by normalization of the kernel) that $k(z, z') \leq 1$ for all $z, z' \in \Omega_* \cup \Gamma(\gamma)$.*

**Lemma F.4.** *Fix some $\gamma > 0$. Suppose that the sequence of calibration coefficients $\left\{\hat{\beta}_t(\delta)\right\}$ used to construct the sets $S_t$ satisfies $\hat{\beta}_t(\delta) \geq \beta_t^{\Omega_* \cup \Gamma(\gamma)}(\delta)$. Let Assumptions F.1-F.2 hold at level $\gamma$ and condition on the calibration event of Theorem F.1 with $A \leftarrow \Omega_* \cup \Gamma(\gamma)$. For $\tau \geq \frac{\exp(\text{logit}(\ell))}{\underline{Z}}$ it holds that $\Omega_t^\tau \subseteq \Omega_* \cup \Gamma(\gamma)$.*

*Proof.* By (44) and the fact that $Z_t \geq \underline{Z}$ it holds that

$$\begin{aligned}
\Omega_t^\tau &\subseteq \{z \in \mathcal{Z} : \mu_t'(z) \geq \log(\tau) + \log(\underline{Z})\} \\
&\subseteq \{z \in \mathcal{Z} : \mu_t'(z) \geq \text{logit}(\ell)\} \\
&\subseteq S_t \cup \{z \in \mathcal{Z} \setminus S_t : d(z, S_t) \leq \gamma\} \\
&\subseteq \Omega_* \cup \{z \in \mathcal{Z} \setminus \Omega_* : d(z, \Omega_*) \leq \gamma\} \\
&= \Omega_* \cup \Gamma(\gamma),
\end{aligned}$$

where the final set inequality follows from Lemma F.2. □

### F.3 Active Diffusion Expansion Core Analysis

By building on the two previous subsections, we can finally analyze the active expansion process of the set $\Omega_t^\tau$. Intuitively, we proceed in two steps: first, we analyze the expansion of the verifier valid set $S_t$ under samples obtained via generative sampling; second, we use Lemma F.3 to transfer this expansion guarantee to the generable set $\Omega_t^\tau$, which is the object we ultimately wish to study.

We adopt the same pre-query indexing convention as in Section F.1. Accordingly, at round $t$ the learner first constructs $\Omega_t^\tau$ and the verifier model $(\mu_t, \sigma_t)$, and then selects the next query point. The sampling scheme used by Algorithm 1 is given by as follows.

**Assumption F.3.** *For some fixed $\alpha \geq 1$, we suppose that the sample points queried by the learner satisfy*

$$z_t \in \Omega_t^\tau \qquad \textit{s.t.} \qquad \sigma_t(z_t) \geq \frac{1}{\alpha} \max_{z \in \Omega_t^\tau} \sigma_t(z). \tag{46}$$

Crucially, the sampling oracle in (46) requires the generative model (e.g., via inference-time techniques) to sample approximate maximizers of $\sigma_t(\cdot)$ over the current generable set $\Omega_t^\tau$.

We ultimately want to show a suitable notion of expansion of the valid set by following the above procedure. As our valid set at any time is defined as an expansion of the valid set at the previous time, we cannot guarantee convergence to the entire subset of $\mathcal{Z}$ which is valid. Instead, we must define some notion of *reachable valid set*. To this end, we consider the tightened one-step reachability operator

$$R_\varepsilon(S) \triangleq \{z \in \mathcal{Z} \mid \exists z' \in S : s(g(z')) - L_s L_g d(z, z') - \varepsilon \geq h\}.$$

The $H$-fold recursive application of this operator is denoted $R_\varepsilon^H(S)$, and its closure as $H \to \infty$ is denoted $\bar{R}_\varepsilon(S)$. Ultimately we show that the valid set discovered by the learner converges to $\bar{R}_0(S_0)$, leading to the following theorem.

**Theorem F.5.** *Fix some $\gamma > 0, \delta > 0$, $\varepsilon > 0$. Let $H$ be a positive integer. Suppose that the sequence of calibration coefficients $\left\{\hat{\beta}_t(\delta)\right\}$ used to construct the sets $S_t$ satisfies $\hat{\beta}_t(\delta) \geq \beta_t^{\Omega_* \cup \Gamma(\gamma)}(\delta)$. Let Assumptions F.1-F.2 hold at level $\gamma$ and condition on the calibration event of Theorem F.1 with $A \leftarrow \Omega_* \cup \Gamma(\gamma)$. Let $\frac{\exp(\mathrm{logit}(\ell))}{Z} \leq \tau \leq \frac{\exp(\mathrm{logit}(h)}{Z}$. Consider sampling with the oracle defined by Assumption F.3 for $T^\star = TH$ steps, with $T$ satisfying*

$$T \geq \frac{8\alpha^2 \hat{\beta}_{HT}(\delta)^2 \gamma_T^{\Omega_* \cup \Gamma(\gamma)}}{\varepsilon^2 \log(1 + (\lambda\kappa)^{-1})}.$$

*Then it holds that $R_\varepsilon^H(S_0) \subseteq \Omega_{T^\star}^\tau$.*

To show the above result, we first prove a few basic inequalities. Our first inequality bounds the uncertainty over the set of valid decisions after $T$ additional algorithm steps.

**Lemma F.6** (Uncertainty Reduction via Local Generative Sampling)**.** *Consider the setting of Lemma F.4. Fix $t \geq 0$ and $T \geq 1$. Under local generative sampling (46), it holds that the epistemic uncertainty at time $t + T$ satisfies*

$$\max_{z \in S_t} \sigma_{t+T}(z) \leq \alpha \sqrt{\frac{2}{\log(1 + (\lambda\kappa)^{-1})}} \sqrt{\frac{\gamma_T^{\Omega_* \cup \Gamma(\gamma)}}{T}}.$$

*Proof.* By monotonicity of the uncertainty score, it holds that

$$\begin{aligned}
\max_{z \in S_t} \sigma_{t+T}(z) &\leq \frac{1}{T} \sum_{n=0}^{T-1} \max_{z \in S_t} \sigma_{t+n}(z) && \text{(uncertainty is monotone)} \\
&\leq \frac{1}{T} \sum_{n=0}^{T-1} \max_{z \in S_{t+n}} \sigma_{t+n}(z) && (S_t \text{ are monotone}) \\
&\leq \frac{1}{T} \sum_{n=0}^{T-1} \max_{z \in \Omega_{t+n}^\tau} \sigma_{t+n}(z) && \text{(Lemma F.3)} \\
&\leq \frac{\alpha}{T} \sum_{n=0}^{T-1} \sigma_{t+n}(z_{t+n}) && \text{(local generative sampling (46))} \\
&\leq \frac{\alpha}{\sqrt{T}} \sqrt{\sum_{n=0}^{T-1} \sigma_{t+n}^2(z_{t+n})} && \text{(Cauchy–Schwarz).}
\end{aligned}$$

It holds that $x \leq \frac{\bar{x}}{\log(1+\bar{x})} \log(1+x)$ for $0 \leq x \leq \bar{x}$. Let $K_{t:t+T-1}$ be the Gram matrix of the queried points $z_t, \ldots, z_{t+T-1}$, let $K_{1:t-1}$ be the Gram matrix of the previously queried points $z_1, \ldots, z_{t-1}$, let $K_{t:t+T-1,\,1:t-1}$ be the corresponding cross-Gram matrix, and define

$$\Sigma_{t:t+T-1|1:t-1} \triangleq K_{t:t+T-1} - K_{t:t+T-1,\,1:t-1}(K_{1:t-1} + \lambda\kappa I_{t-1})^{-1} K_{1:t-1,\,t:t+T-1}.$$

Then the above inequality along with the fact that $k(z, z') \leq 1$ for all $z, z' \in \Omega_* \cup \Gamma(\gamma)$ implies that

$$\begin{aligned}
\max_{z \in S_t} \sigma_{t+T}(z) &\leq \frac{\alpha}{\sqrt{T}} \sqrt{\frac{1}{\log(1 + (\lambda\kappa)^{-1})}} \sqrt{\sum_{n=0}^{T-1} \log\left(1 + \frac{\sigma_{t+n}^2(z_{t+n})}{\lambda\kappa}\right)} \\
&= \frac{\alpha}{\sqrt{T}} \sqrt{\frac{1}{\log(1 + (\lambda\kappa)^{-1})}} \sqrt{\log\det(I_T + (\lambda\kappa)^{-1}\Sigma_{t:t+T-1|1:t-1})} \\
&\leq \frac{\alpha}{\sqrt{T}} \sqrt{\frac{1}{\log(1 + (\lambda\kappa)^{-1})}} \sqrt{\log\det(I_T + (\lambda\kappa)^{-1}K_{t:t+T-1})} \\
&\leq \frac{\alpha}{\sqrt{T}} \sqrt{\frac{1}{\log(1 + (\lambda\kappa)^{-1})}} \sqrt{\max_{z_1,\ldots,z_T \in \Omega_* \cup \Gamma(\gamma)} \log\det(I_T + (\lambda\kappa)^{-1}K_T)} \\
&= \alpha\sqrt{2} \frac{\sqrt{\gamma_T^{\Omega_* \cup \Gamma(\gamma)}}}{\sqrt{T}} \sqrt{\frac{1}{\log(1 + (\lambda\kappa)^{-1})}}.
\end{aligned}$$

Here the last inequality uses Lemma F.4, which implies that all queried points $z_t, \ldots, z_{t+T-1}$ lie in $\Omega_* \cup \Gamma(\gamma)$. □

The next lemma shows that once the uncertainty over the current valid set is sufficiently small, additional local generative sampling expands the valid set according to the one-step reachability operator.

**Lemma F.7** (One-step Reachable Expansion via Local Generative Sampling)**.** *Consider the setting of Lemma F.6. Let $\varepsilon > 0$, $\delta > 0$, $t \geq 0$, and $T \geq 1$. Suppose*

$$T \geq \frac{8\alpha^2 \hat{\beta}_{t+T}(\delta)^2 \gamma_T^{\Omega_* \cup \Gamma(\gamma)}}{\varepsilon^2 \log(1 + (\lambda\kappa)^{-1})}.$$

*Then, it holds that*

$$R_\varepsilon(S_t) \subseteq S_{t+T}.$$

*Proof.* By Lemma F.6, the condition on $T$ implies that

$$\max_{z \in S_t} \sigma_{t+T}(z) \leq \frac{\varepsilon}{2\hat{\beta}_{t+T}}, \tag{47}$$

where we have adopted the shorthand $\hat{\beta}_t$ for $\hat{\beta}_t(\delta)$. Consider any point $z \in R_\varepsilon(S_t)$. By definition, there exists $z' \in S_t$ such that

$$h \leq s(g(z')) - \varepsilon - L_s L_g d(z, z').$$

By the confidence event at round $t + T$,

$$s(g(z')) \leq s(\mu_{t+T}(z')) + \hat{\beta}_{t+T}\sigma_{t+T}(z').$$

Hence

$$\begin{aligned}
h &\leq s(g(z')) - \varepsilon - L_s L_g d(z, z') \\
&\leq s(\mu_{t+T}(z')) + \hat{\beta}_{t+T}\sigma_{t+T}(z') - \varepsilon - L_s L_g d(z, z') \\
&\leq s(\mu_{t+T}(z')) - \hat{\beta}_{t+T}\sigma_{t+T}(z') - L_s L_g d(z, z') \qquad (z' \in S_t, \text{ substitute (47)}).
\end{aligned}$$

Since the sets $S_t$ are monotone, $z' \in S_t$ implies $z' \in S_{t+T-1}$. Therefore there exists $z' \in S_{t+T-1}$ such that

$$h \leq s(\mu_{t+T}(z')) - \hat{\beta}_{t+T}\sigma_{t+T}(z') - L_s L_g d(z, z').$$

By the definition of $S_{t+T}$ in (38), this implies $z \in S_{t+T}$. □

Applying the above result recursively leads to $H$-step expansion of $S_0$ at some timestep $T^\star$. In particular, suppose $T^\star = HT$ for some $T$ satisfying

$$T \geq \frac{8\alpha^2 \hat{\beta}_{HT}(\delta)^2 \gamma_T^{\Omega_* \cup \Gamma(\gamma)}}{\varepsilon^2 \log(1 + (\lambda\kappa)^{-1})}.$$

This implies that the condition of the above lemma holds for each interval of length $T$, since $\gamma_t$ and $\hat{\beta}_t$ are monotonically increasing. Consequently, it holds that

$$R_\varepsilon^H(S_0) \subseteq S_{T^\star}.$$

We can find $T^\star$ large enough to satisfy this condition as long as the complexity term $\hat{\beta}_t^2 \gamma_t^{\Omega_* \cup \Gamma(\tau)}$ grows sublinearly. Ultimately, by Lemma F.3, we have

$$S_{T^\star} \subseteq \Omega_{T^\star}^\tau,$$

which implies that the generable set $\Omega_{T^\star}^\tau$ obtained after running the algorithm is a superset of the valid reachable set, namely

$$R_\varepsilon^H(S_0) \subseteq S_{T^\star} \subseteq \Omega_{T^\star}^\tau. \tag{48}$$

While (48) states that the generable set of the extended diffusion model is a superset of the reachable valid set, it does not clarify whether enough model density has been placed within the valid reachable set $R_\varepsilon^H(S_0) \subseteq S_{T^\star}$ rather than within $\Omega_{T^\star}^\tau \setminus R_\varepsilon^H(S_0)$, which might contain invalid points. To shed light on this question, we next derive a pointwise lower bound on the final density over the generable set.

### F.4 Deriving a Lower Bound on Validity

We now show that samples from the generative model trained after $T^\star$ are valid with high probability.

**Corollary F.8.** *Under the setting of Theorem F.5 and by the construction of the EBM model, it holds that*

$$\mathbb{P}_{z \sim p_{T^\star}}[z \in \Omega_*] \geq \mathsf{vol}(R_\varepsilon^H(S_0)) \frac{\exp(\mathrm{logit}(h))}{\bar{Z}}.$$

*Proof.* It holds by definition of $p_{T^\star}$ that

$$\begin{aligned}
\mathbb{P}_{z \sim p_{T^\star}}[z \in \Omega_*] &= \int_{z \in \mathcal{Z}} \mathbf{1}(\Omega_*)(z) p_{T^\star}(z) dz \\
&= \int_{z \in \mathcal{Z}} \mathbf{1}(\Omega_*)(z) \frac{\exp(\mu'_{T^\star}(z))}{\int_{\mathcal{Z}} \exp(\mu'_{T^\star}(z)) dz} dz \\
&\geq \int_{z \in R_\varepsilon^H(S_0)} \mathbf{1}(\Omega_*)(z) \frac{\exp(\mu'_{T^\star}(z))}{\bar{Z}} dz \\
&= \int_{z \in R_\varepsilon^H(S_0)} \frac{\exp(\mu'_{T^\star}(z))}{\bar{Z}} dz \\
&\geq \int_{z \in R_\varepsilon^H(S_0)} \frac{\exp(\mathrm{logit}(h))}{\bar{Z}} dz,
\end{aligned}$$

where the last equality follows from the fact that $R_\varepsilon^H(S_0) \subseteq \Omega_*$, and the last inequality follows from the fact that $R_\varepsilon^H(S_0) \subseteq S_{T^\star}$ and $\mu'_{T^\star}(z) \geq \mathrm{logit}(h)$ for $z \in S_{T^\star}$. □

### F.5 Proof of design-space coverage corollary

**Assumption F.4** (Fixed representation with nondegenerate Jacobian)**.** *The representation map $\phi : \mathcal{X} \to \mathcal{Z}$ is fixed throughout ADE, is a $C^1$-diffeomorphism onto $\mathcal{Z} = \phi(\mathcal{X})$, and satisfies $|\det J\phi(x)| \geq j_{\min} > 0$ for all $x \in \mathcal{X}$.*

**Corollary 4.2** (Design-space coverage of the induced reachable valid set)**.** *Assume the conditions of Theorem 4.1 and Assumption F.4, with* $j_{\min} := \inf_{x\in\mathcal{X}} |\det J_\phi(x)| > 0$*. Let* $S_0^X := \phi^{-1}(S_0)$ *and* $\tau_X := j_{\min}\tau$*. Then, after the same number* $T^\star$ *of verified samples as in Theorem 4.1, with probability at least* $1-\delta$*,*

$$\big(R_\epsilon^{X,\phi}\big)^H(S_0^X) \subseteq \Omega_{T^\star}^{X,\tau_X}. \tag{20}$$

*Proof.* Condition on the event of Theorem 4.1, which holds with probability at least $1-\delta$. On this event,

$$R_\epsilon^H(S_0) \subseteq \Omega_{T^\star}^{\tau} = \{z \in \mathcal{Z} :\ p'_{T^\star}(z) \geq \tau\}. \tag{49}$$

We first show that for every $A \subseteq \mathcal{X}$,

$$\phi\big(R_\epsilon^{X,\phi}(A)\big) = R_\epsilon\big(\phi(A)\big). \tag{50}$$

Indeed, if $x \in R_\epsilon^{X,\phi}(A)$, then for some $x' \in A$,

$$s\big(g(\phi(x'))\big) - L_s L_g\, d\big(\phi(x), \phi(x')\big) - \epsilon \geq h.$$

Writing $z = \phi(x)$ and $z' = \phi(x')$, we get $z' \in \phi(A)$ and

$$s(g(z')) - L_s L_g\, d(z, z') - \epsilon \geq h,$$

hence $z \in R_\epsilon(\phi(A))$. This proves $\phi(R_\epsilon^{X,\phi}(A)) \subseteq R_\epsilon(\phi(A))$.

Conversely, if $z \in R_\epsilon(\phi(A))$, then for some $z' \in \phi(A)$,

$$s(g(z')) - L_s L_g\, d(z, z') - \epsilon \geq h.$$

Since $\phi$ is bijective, there exist unique $x = \phi^{-1}(z)$ and $x' = \phi^{-1}(z')$ with $x' \in A$. Substituting $z = \phi(x)$ and $z' = \phi(x')$ gives

$$s\big(g(\phi(x'))\big) - L_s L_g\, d\big(\phi(x), \phi(x')\big) - \epsilon \geq h,$$

so $x \in R_\epsilon^{X,\phi}(A)$ and thus $z = \phi(x) \in \phi(R_\epsilon^{X,\phi}(A))$. Therefore (50) holds.

Iterating (50) yields

$$\phi\Big(\big(R_\epsilon^{X,\phi}\big)^H(S_0^X)\Big) = R_\epsilon^H\big(\phi(S_0^X)\big) = R_\epsilon^H(S_0), \tag{51}$$

where we used $S_0^X = \phi^{-1}(S_0)$.

Now let $x \in \big(R_\epsilon^{X,\phi}\big)^H(S_0^X)$. By (51), its image $z := \phi(x)$ belongs to $R_\epsilon^H(S_0)$, hence by (49),

$$p'_{T^\star}(z) \geq \tau. \tag{52}$$

Since $\phi$ is a $C^1$-diffeomorphism, the change-of-variables formula implies that the design-space and representation-space densities satisfy

$$p_1^{\pi_{T^\star}}(x) = p'_{T^\star}(\phi(x))\, |\det J\phi(x)|.$$

Combining this with (52) and the Jacobian lower bound from Assumption F.4 gives

$$p_1^{\pi_{T^\star}}(x) \geq \tau\, j_{\min} = \tau_X.$$

Therefore $x \in \Omega_{T^\star}^{X,\tau_X}$ by (18). Since $x$ was arbitrary, (20) follows. □

### F.6 Full design-space coverage under global reachability

**Corollary F.9** (Full coverage under finite-chain global reachability)**.** *Assume the conditions of Theorem F.5 and Assumption F.4. Let*

$$S_0^X := \phi^{-1}(S_0), \qquad \tau_X := j_{\min}\tau,$$

*as in Corollary 4.2. Suppose that for every design* $x \in \Omega_*$ *there exists a finite chain*

$$x_0, \ldots, x_m \in \Omega_*, \qquad x_0 \in S_0^X, \quad x_m = x, \quad m \leq H,$$

*such that for every* $i = 1, \ldots, m$*,*

$$L_s L_g\, d\big(\phi(x_i), \phi(x_{i-1})\big) + \varepsilon \ \leq\ s\big(g(\phi(x_{i-1}))\big) - h.$$

*Then*

$$\Omega_* \subseteq \Omega_{T^\star}^{X,\tau_X}$$

*with probability at least* $1-\delta$*, after the same number* $T^\star = HT$ *of verified samples as in Theorem F.5.*

*Proof.* Condition on the calibration event used in Theorem F.5, which holds with probability at least $1-\delta$ by Theorem F.1 under the conditions of Theorem F.5. Fix $x \in \Omega_*$, and let $x_0, \dots, x_m$ be the chain given by the assumption. We first prove by induction that

$$x_i \in \big(R_\varepsilon^{X,\phi}\big)^i(S_0^X) \qquad \text{for all } i \le m,$$

with the convention $\big(R_\varepsilon^{X,\phi}\big)^0(S_0^X) = S_0^X$. The base case $i=0$ is immediate since $x_0 \in S_0^X$. Assume $x_{i-1} \in \big(R_\varepsilon^{X,\phi}\big)^{i-1}(S_0^X)$. By the chain condition,

$$s\big(g(\phi(x_{i-1}))\big) - L_s L_g\, d\big(\phi(x_i), \phi(x_{i-1})\big) - \varepsilon \ge h.$$

Therefore, by the definition of $R_\varepsilon^{X,\phi}$ in (19),

$$x_i \in R_\varepsilon^{X,\phi}\Big(\big(R_\varepsilon^{X,\phi}\big)^{i-1}(S_0^X)\Big) = \big(R_\varepsilon^{X,\phi}\big)^i(S_0^X).$$

Thus

$$x = x_m \in \big(R_\varepsilon^{X,\phi}\big)^m(S_0^X).$$

Let $z = \phi(x)$. Iterating the intertwining identity (50) $m$ times gives

$$\phi\left(\big(R_\varepsilon^{X,\phi}\big)^m(S_0^X)\right) = R_\varepsilon^m\big(\phi(S_0^X)\big) = R_\varepsilon^m(S_0),$$

where the last equality uses $S_0^X = \phi^{-1}(S_0)$. Hence

$$z \in R_\varepsilon^m(S_0).$$

We now show that every intermediate exact iterate is covered by the certified set at the corresponding block time. Specifically, for every $r \le H$,

$$R_\varepsilon^r(S_0) \subseteq S_{rT}.$$

The claim holds for $r=0$. If it holds for some $r < H$, then by monotonicity of the operator $R_\varepsilon$ and by Lemma F.7, applied on the block starting at time $rT$,

$$R_\varepsilon^{r+1}(S_0) = R_\varepsilon\big(R_\varepsilon^r(S_0)\big) \subseteq R_\varepsilon(S_{rT}) \subseteq S_{(r+1)T}.$$

Here the sample-complexity condition in Theorem F.5 ensures that Lemma F.7 applies on every block of length $T$. Since $m \le H$, the preceding inclusion and the monotonicity of the certified sets $S_t$, which follows from their recursive definition in (38), imply

$$z \in R_\varepsilon^m(S_0) \subseteq S_{mT} \subseteq S_{HT} = S_{T^\star}.$$

By Lemma F.3,

$$S_{T^\star} \subseteq \Omega_{T^\star}^{\tau}.$$

Therefore $p'_{T^\star}(\phi(x)) \ge \tau$. Finally, by the change-of-variables formula and Assumption F.4,

$$p_1^{\pi_{T^\star}}(x) = p'_{T^\star}(\phi(x))\,|\det J\phi(x)| \ge \tau j_{\min} = \tau_X.$$

Hence $x \in \Omega_{T^\star}^{X,\tau_X}$. Since $x \in \Omega_*$ was arbitrary,

$$\Omega_* \subseteq \Omega_{T^\star}^{X,\tau_X}.$$

□

**Interpretation of reachability.** Reachability is defined through local valid expansions in the learned representation space $\mathcal{Z}$, rather than through distances in the raw design space. The reachable set gives a lower bound on the coverage guaranteed by the analysis, rather than an upper bound on the coverage attainable by ActFlow. In practice, the method may discover an additional valid region and seed further local expansion. However, discovery is not guaranteed for a component that is neither sampled nor reachable from the initial generable set through valid expansion steps.

## G ActFlow for Discrete Diffusion

To adapt ACTFLOW for *discrete diffusion models*, we leverage the algorithm introduced in prior work for adapting a pre-trained discrete diffusion model to an intractable reward-tilted distribution [52]. Given a reward function defined over clean sequences, this method provably tilts a pre-trained discrete diffusion model to the reward-tilted distribution using off-policy reinforcement learning. The full algorithm is given in Alg 2.

### G.1 Discrete Diffusion as Continuous-Time Markov Chains

In contrast to the continuous state space, where generative models are defined by stochastic differential equations (SDEs) or ordinary differential equations (ODEs), generative models in the discrete state space $\mathcal{V} := \{1, \dots V\}$ are defined by **continuous-time Markov chains** (CTMCs). A CTMC is a stochastic process $X_{0:1} := (X_s)_{s\in[0,1]}$ whose probability law is defined by a *generator* $(\boldsymbol{Q}_s)_{s\in[0,1]} \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ defined as:

$$\boldsymbol{Q}_s(x, y) = \lim_{\Delta s\to 0} \frac{1}{\Delta s}(\Pr(X_{s+\Delta s} = y|X_s = x) - \mathbf{1}_{x=y}) \tag{53}$$

which defines the probability of transitioning from state $x \in \mathcal{V}$ to state $y \in \mathcal{V}$ at time $s$.

Masked discrete diffusion models (MDMs) [45, 43, 37, 61] are a class of discrete diffusion models that aim to generate sequences $x_1 \sim p_{\text{data}} \in \mathcal{V}^L$ of length $L$ from a sequence of absorbing *mask tokens* $\boldsymbol{M}$. The generative process is defined as a CTMC that evolves from a prior distribution $p_0$ defined as the Dirac delta of fully masked sequences to the data distribution $p_1 \equiv p_{\text{data}}$. Since each position along the sequence $\ell \in \{1, \dots, L\}$ transitions from a mask token $X^\ell = \boldsymbol{M}$ to a clean token $X^\ell = x^\ell$, the generator can be parameterized as:

$$\boldsymbol{Q}_s(x, x^{\ell\leftarrow v}) = \gamma(s)p^{\theta_t}(\cdot|x)_{\ell,v} \tag{54}$$

where $x^{\ell\leftarrow v}$ denotes the sequence where the $\ell$th token is replaced with state $v \in \mathcal{V}$ and $\gamma(t)$ is the forward noising schedule. To train $p^{\theta_t}(\cdot|x)$ to reconstruct sequences from the data distribution, we can optimize a **denoising cross-entropy (DCE)** loss [45, 43, 37] defined as:

$$\mathcal{L}_{\text{DCE}}(\theta; x_1) := \mathbb{E}_{s\sim\mathcal{U}(0,1)}\left[\frac{1}{s}\mathbb{E}_{p_s(\tilde{x}_s|x_1)}\sum_{\ell:\tilde{x}_s^\ell=\boldsymbol{M}} -\log p^{\theta_t}(\cdot|\tilde{x}_s^\ell)_\ell\right], \quad x_1 \sim p_{\text{data}} \tag{55}$$

where $p_s(\tilde{x}_s|x_1)$ is the distribution of partially masked sequences obtained by masking each position with probability $s$ given a clean sequence $x_1 \sim p_{\text{data}}$.

### G.2 Entropy-Regularized Uncertainty Optimization for Discrete Diffusion

Given a function $\sigma(\cdot) : \mathcal{V}^L \to \mathbb{R}$ that returns the epistemic uncertainty of a sequence $x_1 \in \mathcal{V}^L$ and a pretrained discrete diffusion model that generates the path measure $\mathbb{P}^{\theta_0}$, we define the **uncertainty-tilted path measure** as:

$$\mathbb{P}^\sigma(X_{0:1}) := \frac{1}{Z}\mathbb{P}^{\theta_0}(X_{0:1})\exp\left(\frac{\sigma(X_1)}{\beta}\right), \quad p_1^\sigma(X_1) \propto p_{\text{data}}(X_1)\exp\left(\frac{\sigma(X_1)}{\beta}\right) \tag{56}$$

This uncertainty-tilted path measure coincides with the solution to the **entropy-regularized reward optimization** problem with reward defined as the uncertainty $r(\cdot) := \sigma(\cdot)$ given by:

$$\arg\max_{\theta_{t+1}} \mathbb{E}_{X_{0:1}\sim\mathbb{P}^{\theta_t}}[\sigma(X_1)] - \beta D_{\text{KL}}(\mathbb{P}^{\theta_t}\|\mathbb{P}^{\theta_0}) \tag{57}$$

where $\mathbb{P}^{\theta_t}$ is the CTMC path measure generated from the adapted model with parameters $\theta$ and $\mathbb{P}^{\theta_0}$ is the frozen pre-trained model. $\beta$ is the weight of the KL regularization with the pre-trained model.

### G.3 Uncertainty-Aware Fine-Tuning of Discrete Diffusion

To adapt a pre-trained discrete diffusion model to align with the *uncertainty-tilted distribution* defined in (56), we leverage the off-policy reinforcement learning algorithm introduced in prior work [52],

**Algorithm 2** ACTFLOW for Discrete Diffusion

**Input:** pre-trained model $p^{\theta_0}(\cdot|X_s)$, uncertainty function $\sigma : \mathcal{V}^L \to \mathbb{R}$, number of iterations $T$, number of WDCE repeats $R$
**for** $t = 0, 1, \ldots, T-1$ **do**
  $\{x_1^i, w^\sigma\}_{i=1}^B \leftarrow \texttt{Generate}(p^{\theta_0}, p^{\theta_t})$
  $\mathcal{B} \leftarrow \{x_1^i, w^\sigma\}_{i=1}^B$ ▷ replay buffer
  **for** step in $1, \ldots, N_{\text{step}}$ **do**
    $\{\tilde{x}_s^i, w^\sigma\}_{i=1}^{B\times R} \leftarrow \texttt{ResampleWithMask}(\mathcal{B}; R)$
    Compute $\mathcal{L}_{\text{WDCE}}$ from (58) with $\{\tilde{x}_s^i, w^\sigma\}_{i=1}^{B\times R}$
    Update $\theta_{t+1}$ with $\nabla_\theta \mathcal{L}_{\text{WDCE}}$
  **end for**
**end for**
**return** adapted model $p^{\theta_T}$

which is minimized when $\mathbb{P}^{u_\theta} = \mathbb{P}^\sigma$. The training objective is defined as the **weighted denoising cross-entropy** loss $\mathcal{L}_{\text{WDCE}}$ given by:

$$\mathcal{L}_{\text{WDCE}} := \mathbb{E}_{X_{0:1}\sim\mathbb{P}^\sigma}\left[\mathcal{L}_{\text{DCE}}(\theta; x_1)\right] = \mathbb{E}_{X_{0:1}\sim\mathbb{P}^{\bar{u}}}\Bigg[\underbrace{\frac{\mathrm{d}\mathbb{P}^\sigma}{\mathrm{d}\mathbb{P}^{\bar{\theta}_t}}}_{\exp(w^\sigma(X_1))}\mathcal{L}_{\text{DCE}}(\theta; x_1)\Bigg] \tag{58}$$

where $\mathbb{P}^{\bar{\theta}_t} = \texttt{stopgrad}(\mathbb{P}^{\theta_t})$ is the CTMC path measure generated from the non-gradient-tracking model. We define the importance weight $w^\sigma(X_1) := \log \frac{\mathrm{d}\mathbb{P}^\sigma}{\mathrm{d}\mathbb{P}^{\bar{\theta}_t}}$ as the Radon-Nikodym derivative between the uncertainty-tilted path measure $\mathbb{P}^\sigma$ and the non-gradient-tracking adapted model $\mathbb{P}^{\bar{\theta}_t}$:

$$\log \frac{\mathrm{d}\mathbb{P}^\sigma}{\mathrm{d}\mathbb{P}^{\bar{\theta}_t}}(X_{0:1}) = \underbrace{\frac{\sigma(X_1)}{\beta} + \sum_{s:X_{s+\Delta s}\neq X_s}\sum_{\ell:X^\ell_{s+\Delta s}\neq X^\ell_s}\log\frac{p^{\theta_0}(X^\ell_{s+\Delta s}|X_s)}{p^{\bar{u}}(X^\ell_{s+\Delta s}|X_s)}}_{w^\sigma(X_{0:1})} - \log Z \tag{59}$$

which reweights trajectories $X_{0:1}$ from the current frozen model by their likelihood under the uncertainty-tilted path measure. To optimize this objective, we iterate through the following steps:

(i) Sample $B$ trajectories $X^i_{0:1}$ from the adapted model $\mathbb{P}^{\bar{\theta}_t}$ without gradient tracking while tracking the log-likelihoods of each step $\log p^{\theta_0}(X_{s+\Delta s}|X_s) - \log p^{\bar{\theta}_t}(X_{s+\Delta s}|X_s)$.

(ii) Compute the importance weights $w^\sigma(X^i_{0:1})$ using the log-likelihoods and the uncertainties evaluated on the clean sequences $\sigma(X^i_1)$.

(iii) Store the clean sequences and their weights in a replay buffer $\mathcal{B} \leftarrow \{x_1^i, w^\sigma(x_1^i)\}_{i=1}^B$.

(iv) Resample $R$ partially masked versions of each clean sequence in the buffer $x_1^i$ at different time steps $s \in [0, 1]$ to obtain $\{\tilde{x}_s^i, w^\sigma(x_1^i)\}_{i=1}^{B\times R}$.

(v) Compute $\mathcal{L}_{\text{WDCE}}(\theta)$ from Eq (58) using $\{\tilde{x}_s^i, w^\sigma(x_1^i)\}_{i=1}^{B\times R}$ and update $\theta$ with $\nabla_\theta \mathcal{L}_{\text{WDCE}}$ for $N_{\text{iter}}$ iterations.

After $T$ iterations, the output is the adapted model with path measure $\mathbb{P}^{\theta_T}$ that generates the distribution $p_1^{\theta_T}$ approximating the uncertainty-tilted path measure $\mathbb{P}^\sigma$ and distribution $p_1^\sigma$.

# H Experimental Details

## H.1 Domain-agnostic evaluation metrics for real-world OOD generative modeling.

Standard evaluation criteria for deep generative models, such as Frechet Inception Distance (FID) [24], aim to assess whether the generative model well-approximates the data distribution $p_{\text{data}}$. This is misaligned with OOD generative modeling: successful expansion should increase valid coverage beyond the initial generable region, and may therefore *increase* distributional distance from the pre-trained model. In low-dimensional illustrative experiments, valid coverage can be estimated directly by discretizing the design space, together with validity. In molecular and protein spaces, however, such direct coverage computation is infeasible. We therefore employ several domain-agnostic evaluation metrics for OOD generative modeling, namely: $(i)$ number of fixed-threshold clusters covered by valid samples, to measure model coverage; $(ii)$ Vendi diversity [19], to measure distributional diversity; $(iii)$ FID to quantify divergence from the pre-trained model distribution, which we wish to increase; and validity percentage, to assess whether model expansion is happening over valid regions. In particular, the number of clusters corresponds to the number of hyper-spheres with a fixed data-specific radius, forming a packing over a finite draw of samples of fixed size across methods. This allows to approximately describe the volume covered by the union of hyper-spheres centered at the generated data points, and is an efficient-to-compute proxy metric to assess coverage of the model generable set.

**Task-agnostic continued pre-training.** By task-agnostic, we refer to adaptation that is specific to a design domain, e.g., molecules or proteins, but independent of a particular downstream reward, e.g., binding affinity. Accordingly, the verifier encodes domain-specific validity, and both its choice and the hyperparameters may depend on the domain.

## H.2 Illustrative 2D Experiments

**Overview.** We evaluate ActFlow in a two-dimensional illustrative design space. The base model is a continuous flow model over $\mathbb{R}^2$. The initial model is deliberately misspecified by centering the pretraining data at $(-1.1, 0)$, using $512$ Gaussian samples with standard deviation $0.1$. The valid region is a $3 \times 3$ checkerboard over $[-3.5, 3.5]^2$: a point is valid if it lies in-bounds and falls in a checkerboard cell with even parity. The base model is pretrained for $2500$ steps with Adam, learning rate $10^{-3}$, and batch size $256$. Results are acquired over $20$ seeds and $95\%$ CIs are shown.

**Algorithm configuration.** We run ActFlow for $500$ iterations and $20$ random seeds. At each iteration, ActFlow self-generates $64$ samples. Fine-tuning uses batch size $256$ and $250$ gradient steps per iteration. Evaluation is performed every $50$ iterations using $3000$ samples for evaluation curves. We use $\beta = 1/13$ and GP with RBF kernel with lengthscale $0.08$, and negative-gradient scale $\alpha_t = 0.005$.

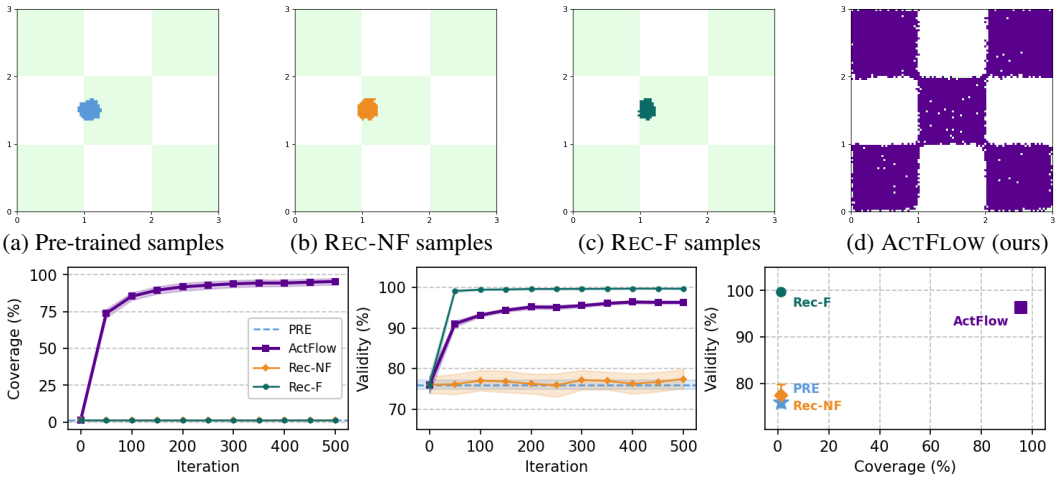
**Uncertainty estimation.** We use a Gaussian process with RBF kernel lengthscale set to $0.08$ and employ flow representation timestep $s = 0.9$

**Validity estimation (i.e., verifier).** The verifier is the deterministic checkerboard validity function. The domain $[-3.5, 3.5]^2$ is partitioned into $3 \times 3$ equal cells. A sample is valid if it lies inside the domain and its cell index $(i, j)$ satisfies $(i + j) \bmod 2 = 0$.

**Coverage metric.** Coverage is measured as generable coverage over the valid region. We draw $3{,}000$ samples from the current model and estimate its generable set on a $100 \times 100$ histogram over $[-3.5, 3.5]^2$. A bin is considered generable if its estimated density is at least $\tau = 0.01$.

**Ablations: no negative gradient in flow matching loss.** We report in Fig. 7 visual results for the illustrative experiments, run with same parameters as in 2, except for $\alpha_t$, which is now set to $0.0$. As one can notice from Fig. 7, the change in this parameters leads to seemingly minimal changes in the results according to the metrics assessed.

**Baselines.** Both baselines use the same number of iterations, samples per iteration, fine-tuning steps, fine-tuning batch size, evaluation schedule, initial invalid model setting, and seeds as ActFlow.




(a) Pre-trained samples (b) REC-NF samples (c) REC-F samples (d) ACTFLOW (ours)

(e) Coverage (f) Validity (g) Coverage-Validity

Figure 7: Results for illustrative experiments with same parameters as in 2, except for $\alpha_t = 0$.

**Hardware and Compute.** Each 2D experiment job requests one RTX 2080 GPU, 16GB memory per CPU, and a four-hour wall-clock limit.

### H.3 Molecular Design: QM9 Experiments

**Overview.** We apply our method to small-molecule generation using FlowMol Gaussian [17] pre-trained on QM9 [41]. Results are acquired over 5 seeds and 95% CIs are shown.

**Algorithm configuration.** We run ACTFLOW for 1,066 iterations, with 66 initial warm-up iterations during which the model is not fine-tuned. Each iteration consists of 64 samples, followed by 500 fine-tuning gradient steps. Fine-tuning uses AdamW with learning rate $10^{-4}$, and batch size 64. Fine-tuning is deferred until 4096 valid samples have been collected, after which the accumulated buffer is used jointly with new guided samples at each iteration. We employ $\beta = 1/10 = 0.1$, and $s = 0.9$.

**Uncertainty estimation.** We use a deep bootstrapped ensemble of 5 MLPs, each with two hidden layers of 100 units, ReLU activations, and 10% dropout. Each ensemble member is trained independently on a 90% bootstrap subsample of the accumulated feature label pairs, using Adam with learning rate $10^{-3}$, for up to 1000 steps. The ensemble standard deviation across members is used as the uncertainty signal.

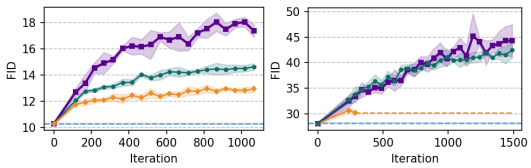
**Validity estimation (i.e., verifier).** A generated molecule is deemed valid if its RDKit-sanitised representation passes valence and bond-order checks and consists of a single connected fragment. Sanitisation is preceded by an MMFF geometry relaxation [23].

**Coverage metric.** Coverage is measured as the number of distinct molecular clusters obtained, computed via greedy sphere exclusion on Morgan fingerprints (radius 2, 2048 bits) using Tanimoto similarity with threshold $\tau = 0.85$: a candidate is added as a new cluster centre if its Tanimoto similarity to all previously selected centres is below $\tau$. This is applied independently to the 500 valid molecules in each evaluation batch.

**Diversity metric.** Vendi score [19] is computed on the same 500 valid molecules per evaluation. The kernel matrix $K$ is the pairwise Tanimoto similarity over 2048-bit Morgan fingerprints (radius 2).

**Baselines.** Both baselines use the same number of iterations, samples per iteration, fine-tuning steps, fine-tuning batch size, evaluation schedule, initial invalid model setting, and seeds as ACTFLOW.

**FID** We report in Fig. 8 FID results over iterates for QM9.




(a) QM9 FID (b) GEOM-Drugs FID

Figure 8: FID results for QM9 and GEOM-Drugs

**Hardware and Compute.** Each run used a single NVIDIA RTX 4090 GPU with 4 CPU cores and 32 GB of system memory, allocated for up to 120 h.

### H.4 Molecular Design: GEOM-Drugs Experiments

**Overview.** We apply our method to small-molecule generation using FlowMol Gaussian [17] pre-trained on GEOM-Drugs [4], a dataset of ∼300 k drug-like organic molecules with energy-annotated conformers. Results are acquired over 3 seeds and 95% CIs are shown.

**Algorithm configuration.** We run ACTFLOW for 1,500 iterations, with 190 initial warm-up iterations during which the model is not fine-tuned. Each iteration consists of 64 samples, followed by 2000 fine-tuning gradient steps. Fine-tuning uses AdamW with learning rate $10^{-4}$, and effective batch size 64. Fine-tuning is deferred until 4096 valid samples have been collected. We employ $\beta = 1/7 \approx 0.14$, and $s = 0.8$.

**Uncertainty estimation.** We use a deep bootstrapped ensemble of 5 MLPs, each with two hidden layers of 100 units, ReLU activations, and 10% dropout. Each ensemble member is trained independently on a 90% bootstrap subsample of the accumulated feature–label pairs, using Adam with learning rate $10^{-3}$, for up to 1000 steps. The ensemble standard deviation across members is used as the uncertainty signal.

**Validity estimation (i.e., verifier).** A generated molecule is deemed valid if its RDKit-sanitised representation passes valence and bond-order checks and consists of a single connected fragment. Sanitisation is preceded by an MMFF geometry relaxation [23].

**Coverage metric.** Coverage is measured as the number of distinct molecular clusters obtained, computed via greedy sphere exclusion on Morgan fingerprints (radius 2, 2048 bits) using Tanimoto similarity with threshold $\tau = 0.85$: a candidate is added as a new cluster centre if its Tanimoto similarity to all previously selected centres is below $\tau$. This is applied independently to the 500 valid molecules in each evaluation batch.

**Diversity metric.** Vendi score [19] is computed on the same 500 valid molecules per evaluation. The kernel matrix $K$ is the pairwise Tanimoto similarity over 2048-bit Morgan fingerprints (radius 2).

**Baselines.** Both baselines use the same number of iterations, samples per iteration, fine-tuning steps, fine-tuning batch size, evaluation schedule, initial invalid model setting, and seeds as ACTFLOW.

**FID** We report in Fig. 8 FID results over iterates for GEOM-Drugs.

**Hardware and Compute.** Each run used a single NVIDIA RTX 4090 GPU with 4 CPU cores and 128 GB of system memory, allocated for up to 120 h.

### H.5 Therapeutic Peptide Design Experiments

**Overview.** We apply ACTFLOW for therapeutic peptide design by adapting the pre-trained discrete diffusion peptide SMILES generator from PepTune [52] trained on 11 million peptide SMILES. The Simplified Molecular-Input Line-Entry System (SMILES) [58] representation enables the generation of non-natural amino acids containing diverse chemical backbone and side-chain modifications, and cyclic modifications, significantly expanding the design space of therapeutic peptides over the

standard 20 natural amino acids. The tokenization uses the SMILES Pair Encoding (SPE) tokenization scheme [30] from PeptideCLM [18] with vocabulary size 586, including 5 special tokens. Results are acquired over 5 seeds and 95% CIs are shown.

**Algorithm configuration.** We run ACTFLOW for 12 rounds following Alg 2, with an initial pool of 1000 peptide sequences generated from the pretrained model and 100 iterations per round. At each round, we draw 100 sequences from the pool, score them with the verifier and Gaussian process uncertainty model, update the policy with the WDCE loss defined in Apx G.3 (16 replicates per sequence), and refresh the training pool by sampling from the updated policy. We train with Adam at a learning rate of $1 \times 10^{-4}$ and batch size 100. The reward is scaled by a parameter $\beta = 0.005$.

**Uncertainty estimation.** Sequence-level uncertainty is obtained from a Gaussian process (GP) fit on top of the diffusion model's RoFormer encoder. For each peptide, we extract attention-pooled, $L_2$-normalised hidden states from the final transformer layer (768-dimensions) and treat the GP conditional posterior variance given the rest of the batch and the GP training set as the uncertainty signal. The GP uses an RBF kernel whose length scale is initialised from the mean pairwise embedding distance of 50 samples from the pretrained model. The GP's posterior is refit at each round (100 iterations) on all sampled sequences. Only sequences that pass the verifier enter the GP buffer.

**Validity estimation (i.e., verifier).** Generated SMILES are first parsed with RDKit. Those that yield a valid Mol object are then decoded with the `SMILES2PEPTIDE` verifier [52], which converts the SMILES into a sequence of natural and non-natural amino acids split on their peptide bonds.

**Coverage metric.** Coverage is measured as the number of distinct neighborhoods occupied by the $N$=1000 valid generated peptides in PeptideCLM [18] embedding space. Each peptide is embedded by mean-pooling the final-layer hidden states of the pretrained PeptideCLM RoFormer over non-padding tokens; the resulting vectors are $L_2$-normalised so that pairwise dissimilarity reduces to the cosine distance $d(x, x') = 1 - \langle x, x' \rangle$. We then apply sphere-exclusion clustering. Iterating through the sequences in arbitrary order, each peptide becomes either a new cluster center if its cosine distance to every existing center is at least $\tau$, or is otherwise assigned to the nearest center. We report the number of clusters at $\tau = 0.10$. Reported metrics are computed after 1200 iterations (12 rounds).

**Diversity metric.** We report the Vendi score [19] on the $L_2$-normalised PeptideCLM embeddings under an RBF kernel $K(x, x') = \exp\left(-\|x - x'\|^2/(2\sigma^2)\right)$ with length-scale $\sigma = 0.5$. Reported metrics are computed after 1200 iterations (12 rounds).

**Fréchet Inception Distance.** To quantify how far each fine-tuned generator drifts from the pretrained distribution of peptides, we report the Fréchet Inception Distance (FID) [24] between two distributions: the 1000 valid peptides generated by the model under evaluation and a matched reference set of 1000 valid peptides sampled from the frozen pretrained checkpoint. The FID score is lower when closer to the pretrained distribution. We compute the score using 2048-bit Morgan fingerprints (radius 2) to evaluate drift in the chemical fingerprint space. Reported metrics are computed after 1200 iterations (12 rounds).

**Baselines.** We compare against three baselines: (1) the pretrained model [52], (2) REC-NF where the policy is updated on its own samples with no uncertainty tilting (uniform weights on WDCE loss), (3) REC-F where the policy is updated on its own samples filtered to retain only valid peptides classified by the verifier. We hold hyperparameters fixed across all baselines.

**Hardware and Compute.** Each mode and seed run is trained on a single NVIDIA B200 GPU with 8 CPU cores and 80 GB of system RAM. A full 100-iteration run fits within a 48-hour wallclock budget per GPU.

### H.6 Protein Sequence Design Experiments

**Overview.** We apply our method to protein sequence design using a continuous ESM diffusion model from SGPO [59]. The base model is a continuous-space denoising network operating over

ESM token-probability vectors of dimension equal to the vocabulary size (31 tokens, including the 20 standard amino acids and special tokens), pre-trained on the CreiLOV fluorescence dataset [10]. CreiLOV is a 119-residue fluorescent protein; the dataset contains experimentally measured fluorescence fitness values for sequence variants. The diffusion process uses a cosine noise schedule, where $\alpha_t = \cos\left(\frac{(1-t)\pi}{2}\right)$ and $\beta_t = \sqrt{1-\alpha_t^2}$, so that $t = 1$ corresponds to data and $t = 0$ to pure noise.

**Algorithm configuration.** We run ACTFLOW for 512 iterations, each consisting of 64 samples, followed by 1000 fine-tuning gradient steps on the accumulated valid samples. Fine-tuning uses AdamW with learning rate $10^{-4}$, batch size 64, and no weight decay. Fine-tuning is deferred until 4096 valid samples have been collected (warm-up period), after which the accumulated buffer is used jointly with new guided samples at each iteration. We use $\beta = 1/50$. Features for the uncertainty estimator are extracted from the encoder of the ESM network at flow representation timestep $s = 0.8$, mean-pooled over the sequence dimension.

**Uncertainty estimation.** We use a deep ensemble of 5 MLPs, each with two hidden layers of 100 units, ReLU activations, and 10% dropout. Each ensemble member is trained independently on a 90% bootstrap subsample of the accumulated feature-label pairs, using Adam with learning rate $10^{-3}$, for up to 1000 steps. The ensemble standard deviation across members is used as the uncertainty signal.

**Validity estimation (i.e., verifier).** A generated sequence is deemed valid if its mean predicted local distance difference test (pLDDT), computed by ESMFold [31], exceeds a threshold of 65. ESMFold is run in batches of 32 sequences.

**Coverage metric.** Coverage is measured as the number of distinct sequence clusters accumulated across all 512 iterations, computed via greedy sphere exclusion on sequence identity with threshold $\tau = 0.35$: a candidate is added as a new cluster center if its sequence identity to all previously selected center is below $\tau$.

**Diversity metric.** Vendi score [19] is computed on token-level ESM embeddings. For each generated sequence, we compute its embedding as the probability vector over the vocabulary projected through the ESM token embedding table ($L_2$-normalised and scaled by $\sqrt{d_{\text{model}}}$), then mean-pooled over the sequence length. Vendi score is computed using an RBF kernel with lengthscale $\ell = 2.0$ applied to these mean-pooled embeddings.

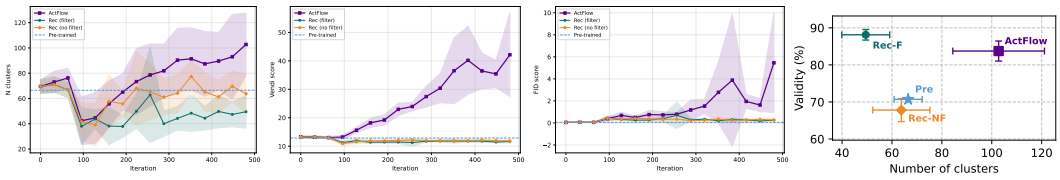
**Ablation: feature timestep.** We ablated the fine-tuning flow representation time-steps $t \in \{0.5, 0.8, 0.9, 0.95\}$ in an alternative configuration with 1024 valid samples are initially queried. Among the feature timestep variants, $t = 0.95$ performed worst in terms of coverage (12 clusters at $\lambda = 50$, Vendi $= 18.6$), showing that extracting features close to the data level might be significantly sub-optimal.

**Baselines.** REC-F runs the same fine-tuning loop without uncertainty-guided sampling, training only on valid samples (pLDDT $> 65$). REC-NF additionally disables the validity filter during fine-tuning, training on all generated samples regardless of pLDDT. Both baselines use identical hyperparameters (fine-tuning steps, learning rate, batch size, warm-up threshold) to ACTFLOW, differing only in whether uncertainty guidance and validity filtering are applied.

**Hardware and Compute.** Each run used a single RTX 4090 GPU for 24 hours and 256 GB of memory.

### H.7 Sensitivity to regularization, representation timestep, and negative feedback

We assess the sensitivity of ACTFLOW to the prior regularization parameter $\beta$, the flow representation timestep $s$, and the strength of the negative-gradient update on invalid samples, denoted by `neg`. For small molecules, we vary one parameter at a time. For protein sequences, we jointly vary $s$ and `neg`. These experiments use shorter adaptation budgets than the main evaluation.




(a) CreiLOV Coverage (b) CreiLOV Diversity (c) CreiLOV FID (d) Coverage-Validity

Figure 9: Results of protein sequence design experiments over iterations (Figs 9a - 9c), and diversity-validity tradeoff at final iteration (Fig 9d). ACTFLOW significantly outperforms REC-NF and REC-F in all diversity metrics (FID, Vendi, number of clusters), while maintaining a competitive validity.

| Parameter varied | Coverage | Validity |
|---|---|---|
| $\beta = 0.2$ | 72.5 | 0.951 |
| $\beta = 0.05$ | 72.5 | 0.946 |
| $s = 0.5$ | 46.5 | 0.926 |
| $s = 0.8$ | 46.5 | 0.932 |
| `neg=0.0001` | 77.0 | 0.949 |
| `neg=0.0005` | 61.0 | 0.931 |

Table 3: Small-molecule sensitivity results after 400 iterations. We report means over two seeds. Validity is reported as a fraction.

### H.8 Structural validity with PoseBusters

We further assess small-molecule structural validity using PoseBusters [8]. We consider two settings. First, we incorporate PoseBusters into the verifier alongside RDKit, so that structural validity informs adaptation. Second, we retain the RDKit verifier during adaptation and use PoseBusters only for evaluation. The latter assesses structural validity under a criterion not used by the adaptation procedure.

| neg | $s$ | Coverage | Validity |
|---|---|---|---|
| Pre-trained | | $71.75 \pm 4.18$ | $0.708 \pm 0.013$ |
| 0.01 | 0.2 | $145.75 \pm 71.26$ | $0.791 \pm 0.081$ |
| 0.01 | 0.5 | $112.00 \pm 35.46$ | $0.819 \pm 0.035$ |
| 0.01 | 0.95 | $81.25 \pm 52.63$ | $0.872 \pm 0.079$ |
| 0 | 0.2 | $139.75 \pm 34.43$ | $0.734 \pm 0.045$ |
| 0 | 0.5 | $94.00 \pm 15.86$ | $0.831 \pm 0.061$ |
| 0 | 0.95 | $109.25 \pm 40.59$ | $0.818 \pm 0.076$ |

Table 4: Protein-sequence sensitivity results after 480 iterations. We report means and 95% confidence intervals over four seeds. Validity is reported as a fraction.

| Method | Coverage | Validity (%) |
|---|---|---|
| *RDKit and PoseBusters in the verifier* | | |
| Pre-trained | $27.0 \pm 0.0$ | $91.6 \pm 0.0$ |
| ActFlow | $81.5 \pm 69.9$ | $92.6 \pm 1.3$ |
| Rec-F | $45.0 \pm 0.0$ | $92.8 \pm 6.4$ |
| Rec-NF | $43.5 \pm 31.8$ | $18.6 \pm 47.3$ |
| *PoseBusters only for evaluation* | | |
| ActFlow | $75.40 \pm 3.37$ | $88.69 \pm 1.27$ |
| Rec-F | $45.00 \pm 2.77$ | $92.77 \pm 1.25$ |
| Rec-NF | $42.40 \pm 2.53$ | $19.44 \pm 3.27$ |

Table 5: Small-molecule structural-validity results. Top: RDKit and PoseBusters are used in the verifier, with results reported after 400 iterations. Bottom: adaptation uses RDKit alone, and PoseBusters is used only for evaluation.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The theoretical and experimental claims made within abstract and introduction are properly supported by evidence, in the form of theoretical and experimental results, within the rest of the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The key limitation of this work is mentioned within the Conclusion section, while theoretical limitations of our analysis are reported in the form of informal assumptions within the theory section. The algorithm proposed employs standard procedures and therefore there is not novelty or limitation specific to this approach in terms of computational complexity and efficiency.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide informal assumptions within the main paper, and formal assumptions within the Appendix, alongside complete derivations.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We wrote an Experimental Details section within the appendix reporting implementation and compute details for the algorithm configurations leading to the results shown. We believe that this data, in combination with the pseudocode provided within the main text, is sufficient to reproduce the main experimental results of the paper as mentioned.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We plan to release code, but at the current time we are not sharing code within this submission, primarily due to lack of time needed for code organization and preparation.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We present such details primarily within the Experimental Details section within the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We show $95\%$ confidence intervals for all experimental results, as detailed within the Experimental Details section within the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details for each experiment in the *Hardware and Compute* paragraphs within the Experimental Details section in the Appendix.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewer the code of ethics and our research conforms with it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: We do not discuss societal impact of this work, as this work is primarily foundational in nature and we are not aware of specific use-cases relevant to discuss.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [N/A]

Justification: The paper does not use existing assets with licensed ownership.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.