Property Adherent Molecular Generation with Constrained Discrete Diffusion

Michael Cardei*
University of Virginia
ntr2rm@virginia.edu

Jacob K. Christopher*
University of Virginia
csk4sr@virginia.edu

Thomas Hartvigsen
University of Virginia
hartvigsen@virginia.edu

Bhavya Kailkhura
Lawrence Livermore National Laboratory
kailkhura1@llnl.gov

Ferdinando Fioretto†
University of Virginia
fioretto@virginia.edu

Abstract

Discrete diffusion models are a class of generative models that construct sequences by progressively denoising samples from a categorical noise distribution. In life science setting, such as molecular strings (SMILES) and other biological sequence design settings, these models have emerged as a promising alternative to autoregressive architectures, presenting an opportunity to enforce sequence-level constraints, a capability that existing left-to-right sequence design cannot natively provide. This paper capitalizes on this opportunity by introducing *Constrained Discrete Diffusion* (CDD), a novel integration of differentiable constraint optimization within the diffusion process to ensure adherence to biosafety policies and design properties during generation. Unlike conventional generators that often rely on post-hoc filtering or model retraining for controllable generation, CDD directly imposes constraints into the discrete diffusion sampling process, resulting in a training-free and effective approach. Experiments in property adherence molecular design, toxicity-bounded generation, and novelty enforcement demonstrate that CDD achieves *zero constraint violations* in a diverse array of tasks outperforming auto-regressive and existing discrete diffusion approaches.

1 Introduction

Many scientific problems admit a natural representation as the generation of a discrete sequence from a finite alphabet. Examples range from molecular SMILES strings to linearized chemical procedures [1, 2]. While discrete sequence foundation models and related sequence transformers have recently accelerated discovery by proposing candidate sequences with desirable attributes, their autoregressive sampling mechanism produces tokens sequentially, hindering the ability to provide a native mechanism to ensure constrained feasibility. When these constraints are not satisfied, the generated outputs may be unreliable and ineffective in real-world applications. For example, an invalid atom in a SMILES string can render a synthesized compound meaningless or even dangerous; similarly an overlooked volume unit in an autonomous laboratory protocol can trigger unsafe reactions.

To limit such risks, bio-generative models are often deployed with a variety of guardrails. These include soft alignment through property based finetuning, rejection sampling, or heuristic post-processing such as SMILES sanitization [3, 4]. However, these methods do not offer provable

*Equal contribution.
†Contact author.

Biosecurity Safeguards for Generative AI Workshop, NeurIPS 2025

compliance or guarantees, as they reduce but do not provably eliminate the generation of outputs that violate property adherence or bio-safety thresholds [5]. This is exacerbated by the auto-regressive nature of common generative models, which generate sequences one token at a time, making it difficult to enforce constraints at the sequence level.

In contrast, discrete diffusion models offer a compelling alternative generative mechanism [6–9]. They refine a fully corrupted sequence by iteratively denoising the entire sample, and have shown strong performance across a variety of tasks, molecular design, program synthesis, and text generation [7, 10, 11]. Since each step exposes a global view of the partial sequence, it creates a natural opportunity to impose sequence-level structure. This work introduces **Constrained Discrete Diffusion** (CDD), a framework that capitalizes on this property by coupling discrete diffusion with a differentiable projection operator.

Given a corrupted sequence at a particular diffusion step, the projection searches for a new candidate that stays close to the model's current score distribution, preserves entropy-based diversity, and simultaneously satisfies all user-defined constraints before the next denoising update. Because the introduced projection acts only at sampling time, the method is *training-free* and needs neither model retraining nor post hoc filtering. Its stochastic formulation also preserves the generative diversity, which is crucial for exploratory scientific design, as will be shown in the empirical analysis.

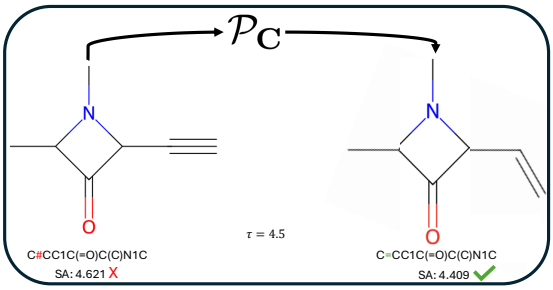

Figure 1: Synthetic accessibility property enforcement via projection operator $\mathcal{P}_{\mathbf{C}}$.

Specifically, CDD is evaluated on three representative molecular generation use cases: (i) property adherence for molecular sequence generation, (ii) safety-driven toxic sequence prevention, and (iii) guaranteeing molecular novelty. The experimental results demonstrate *zero threshold violations* for property adherence and up to a *203.4% increase* in novel molecule generation.

Contributions. This paper provides the following contributions:

1. It introduces Constrained Discrete Diffusion (CDD), a novel framework that integrates discrete diffusion models with a differentiable projection operator, enforcing global sequence-level constraints directly within the diffusion sampling process.
2. It formulates this projection operator by solving a Lagrangian dual program at each denoising step, making the constrained optimization tractable and effective for guiding sequence generation.
3. Through three extensive experiments, we demonstrate that CDD achieves state-of-the-art constraint adherence (zero violations across all settings) while preserving high sample quality in terms of validity, novelty, and property adherence.

2 Related Work

Recent efforts in controllable sequence generation have largely focused on constrained decoding, where the output is dynamically restricted to adhere to predetermined syntactic or grammatical rules. Approaches in this area modify the decoding process to prune the vocabulary, ensuring that only tokens compliant with a formal grammar are considered [12], or employ external modules to filter outputs when direct access to the model's logits is limited [13]. Other methods have also adapted beam search to encourage the presence or absence of specific tokens [14, 15]. Although these techniques effectively guide the generation process, they depend on augmented sampling heuristics which encourage, but frequently fail to provide, satisfaction of even simple constraint sets [16]. In biological sequence generation, structure can be enforced by constraining the representation itself. For molecules, for example, Junction Tree VAE decomposes molecules into valid substructures to improve validity and synthesizability [17]. However, such approaches typically require domain-specific encoders/decoders and still provide no formal guarantee that downstream property or safety thresholds are always satisfied.

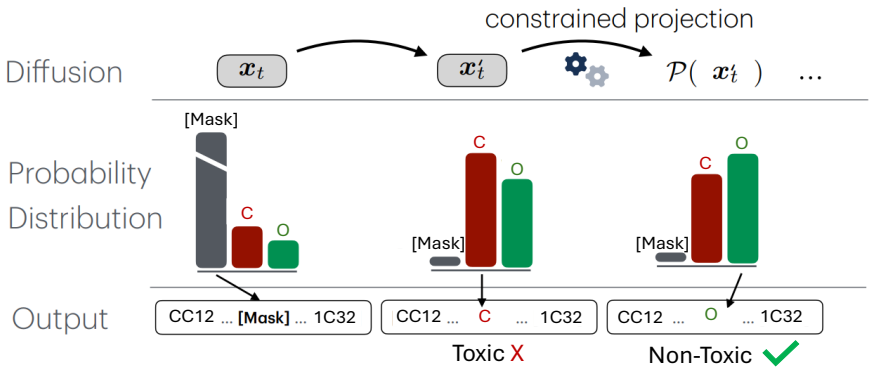

Figure 2: Illustration of CDD's projection step embedded throughout the sampling process.

More recently, several gradient-based sampling frameworks have been proposed to impose constraints on token sequences [18]. Building on this approach, Amini et al. [19] improve generation quality via structured Hamiltonian Monte Carlo. However, these methods lack mechanisms for enforcing hard constraints, and are limited to soft attribute control. Guided discrete diffusion methods for protein design, for example, *Protein Design with Guided Discrete Diffusion* [4] applies conditional guidance during sampling to steer sequences toward desired structural or functional targets and [8] introduces classifier-free and classifier-based guidance have been adapted to discrete diffusion for property steering. Compared to these guidance methods, our presented method solves a different mathematical problem and has a different behavior and guarantees.

We compare against these methods directly, and show that while these methods can shift output distributions, they fail to reliably enforce structural or semantic constraints.

3 Preliminaries: Discrete Diffusion Models

While diffusion models were originally developed for continuous data, they have recently been extended to discrete domains, enabling non-autoregressive generation [7, 9, 20, 21]. In contrast to autoregressive models which predict tokens one by one, *discrete diffusion* methods generate entire sequences in parallel by first corrupting sequences through a forward noising process and then iteratively reconstructing them with a learned reverse process. *This is a key enabler recognized by this work, which exploits this modus operandi to impose global constraints while simultaneously maintaining high fidelity*.

Let $\boldsymbol{x}_0 = (\boldsymbol{x}_0^1, \ldots, \boldsymbol{x}_0^L)$ denote an input sequence of size L, where each token $\boldsymbol{x}_0^i \in \mathcal{V}$ is represented as a one-hot vector over a vocabulary $\mathcal{V}$ of N distinct tokens. In discrete diffusion models, the forward process produces, at time $t \in [0, T]$, a sequence of values $\boldsymbol{x}_t \in \mathcal{V}^L$ that parametrizes probability distributions for each token $\boldsymbol{x}_t^i$ in the sequence. We denote the corresponding sequence of predicted tokens by $\boldsymbol{x}_t^\star = \arg\max(\boldsymbol{x}_t)$ where the $\arg\max$ operator is applied to every member $\boldsymbol{x}_t^i$ of the sequence $\boldsymbol{x}_t$. The diffusion process specifies a *forward transition*, defined as:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathrm{Cat}\big(\boldsymbol{x}_t; \alpha_t \boldsymbol{x}_0 + (1 - \alpha_t)\, \nu\big), \tag{1}$$

where α_t is a schedule that decreases over time, $\mathrm{Cat}(\cdot; p)$ is the categorical distribution over probability vector $p \in \Delta^N$, and ν is a fixed reference distribution that specifies the type of corruption applied to each token. For example, *Uniform Diffusion Language Models* (UDLM) [8], uses ν as the uniform distribution over the vocabulary. This instantiation allows tokens to be re-perturbed in later time steps.

In the *reverse process*, $\boldsymbol{x}_T$ is initialized from ν. For each position i in the sequence, while $\boldsymbol{x}_t^i = \nu$, the denoiser parameterizes the transitions to clean data, $\boldsymbol{x}_0$, and the probability of $\boldsymbol{x}_t^i = \nu$ is discounted as $t \to 0$. Once $\boldsymbol{x}_t \neq \nu$, the probability vector transitions into a one-hot vector, concentrating all

weight exclusively on the predicted token.

$$q(\boldsymbol{x}_s \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \begin{cases} \mathrm{Cat}\big(\boldsymbol{x}_s; \boldsymbol{x}_t\big), & \text{if } \boldsymbol{x}_t \neq \boldsymbol{\nu}, \\ \mathrm{Cat}\Big(\boldsymbol{x}_s; \frac{(1-\alpha_s)\,\boldsymbol{\nu} + (\alpha_s - \alpha_t)\mathbf{x}_\theta(\boldsymbol{x}_t, t)}{1-\alpha_t}\Big), & \text{if } \boldsymbol{x}_t = \boldsymbol{\nu}. \end{cases} \tag{2}$$

Equation (2) formalizes this reverse process parameterization, where $\mathbf{x}_\theta(\boldsymbol{x}_t, t)$ denotes the trained denoiser and s is the subsequent timestep in the continuous time reverse process and $0 \leq s < t \leq T$.

4 Constrained Discrete Diffusion (CDD)

Projected diffusion sampling. We begin by introducing a perspective of the discrete diffusion reverse process that motivates our approach. Prior work has shown that *continuous* score-based diffusion sampling processes can be framed as an sequential optimization procedure [22]. Score-based parameterizations enable this framing as the model learns to predict the gradients of the data density function, $\mathbf{x}_\theta(\boldsymbol{x}_t, t) \approx \nabla \log q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ through Score Matching, and these gradients can then be applied to optimize $\boldsymbol{x}_t$ to the data distribution $q_t(\boldsymbol{x}_t)$. This is typically conducted through a Langevin dynamics algorithms [23], which is used either directly [24] or through predictor-corrector Euler-discretizations [25]. In both cases, a series of M steps of Langevin dynamics are taken for a fixed distribution $q_t(\boldsymbol{x}_t)$:

$$\boldsymbol{x}'_{t(\ell)} \xleftarrow{\text{update } (\ell)} \boldsymbol{x}_{t(\ell)} + \gamma_t \nabla_{\boldsymbol{x}_{t(\ell)}} \log q_t(\boldsymbol{x}_{t(\ell)} | \boldsymbol{x}_0) + \sqrt{2\gamma_t}\boldsymbol{\epsilon} \qquad (\text{for } \ell = 1 \ldots M), \tag{3}$$

where γ_t is the step size, ϵ is added noise, and $\log q_t(\boldsymbol{x}_t)$ is the density function of the *learned* distribution at time step t. Note that while *annealing* is used to improve convergence, it is applied only after every M iterations. At that point, the sample is updated ($\boldsymbol{x}_{t(M)} \to \boldsymbol{x}_{t-1(1)}$) and the model transitions to the next distribution, $q_{t-1}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)$, however, the step size and the distribution $q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ remain stationary throughout these M iterations.

Discrete diffusion models can leverage a discrete generalization of the score function, referred to as the *Concrete score*, to approximate the gradient of the probability density function [7, 9, 26]. Concrete Score Matching provides an approach mirroring continuous Score Matching in that the estimated gradients of the probability density function are used to guide the sample to high density regions of the target distribution. While not always explicitly framed as Concrete Score Matching [9, 26], the denoiser often implicitly models the score function, as supported by theoretical results in [7] demonstrating its equivalence in simplified formulations ($\nabla \log q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \approx \langle \mathbf{x}_\theta(\boldsymbol{x}_t, t), \mathbf{y} \rangle \;\; \forall \mathbf{y} = 1, \ldots, N$, in this case). This enables the use of Langevin-based samplers for discrete diffusion, as commonly employed, e.g., in [26].

Effectively, under some regularity conditions, Langevin dynamics will converge towards a stationary point. As shown by Xu et al., Langevin dynamics acts as an "almost-minimizer" on the optimized function, which in this case will be the negative probability density function, converging within a fixed bound of the optimum. Hence, for each series of steps with a stationary density function, within this fixed bound, the sampling procedure can be framed as the optimization problem,

$$\underset{\boldsymbol{x_t}}{\text{minimize}} \sum_{t=T}^{1} -\log q_t(\boldsymbol{x}_t | \boldsymbol{x}_0) \quad \text{s.t.} \;\; \boldsymbol{x}_t \in \mathbf{C}. \tag{4}$$

Here, Equation (4) extends the representation from an unconstrained optimization problem to a constrained version by introducing a constraint set $\mathbf{C}$. However, while iteratively applying the denoiser enables sampling from the posterior distribution, it does not natively incorporate externally defined constraints, or even implicit ones, given the stochastic nature of the process. Previous work for continuous modalities has proposed enforcing $\boldsymbol{x}_t \in \mathbf{C}$ by applying a Euclidean projection after each denoising step [22], which is natural for continuous modalities that fall in a real space, but this is misaligned when applied to discrete diffusion which operates over a probability simplex $\boldsymbol{x}_t \in \Delta^N$.

To address this, we introduce a projection operator that minimizes the *Kullback-Leibler (KL) divergence* between the projected and original probability distributions, as opposed to a Euclidean distance metric. Given the model's predicted probabilities, the projection is defined as:

$$\boldsymbol{x}_s = \boldsymbol{x}_{t(\ell+1)} = \mathcal{P}_{\mathbf{C}}\big(\boldsymbol{x}'_{t(\ell)}\big) \stackrel{\text{def}}{=} \arg\min_{\boldsymbol{y}} D_{\mathrm{KL}}\left(\boldsymbol{x}'_{t(\ell)} \| \boldsymbol{y}\right) \quad \text{s.t.} \;\; \arg\max(\boldsymbol{y}) \in \mathbf{C}. \tag{5}$$

The integration of this projection operator ensures that $\boldsymbol{x}_s$ is the "nearest sample" that falls within the feasible subdistribution according to a distance metric defined over probability distributions. This ensures the denoising trajectory remains within the allowable set when $\mathbf{C}$ is convex, enabling effective navigation along its boundary or interior toward an optimal solution (illustrated in Figure 2). Next, we show how this projection operator can be formulated as a subproblem within the sampling procedure and efficiently solved using gradient-based methods. Moreover, while convergence guarantees are available for convex constraint sets, as discussed below, Section 5 demonstrates how this technique can effectively handles highly non-linear constraints, including toxicity mitigation and molecular generation, achieving zero violations across all cases.

Importantly, while state transitions are imposed on the token probability distributions, constraint satisfaction is evaluated on the *decoded sequence*. Indeed, the constraints are formulated such that the $\arg\max$ of the projected sequence $\boldsymbol{y}$ (referred to as $\boldsymbol{y}^\star$), must satisfy sentence level criteria $\boldsymbol{y}^\star \in \mathbf{C}$.[3] However, this use of a non-differentiable $\arg\max$ operation also poses challenges to the resolution of the projection, which relies on gradient-based optimization.

Differentiable projection. To address the non-differentiability of the $\arg\max$ function, we apply a Gumbel-Softmax relaxation $\tilde{\phi}$, which approximates discrete token probabilities with continuous values [28] as

$$\tilde{\phi}(\boldsymbol{x}_t^i)(v) = \frac{\exp\left(\frac{\log \boldsymbol{x}_t^i(v) + \xi_v}{\mathrm{T}_{\text{sample}}}\right)}{\sum_{v'=1}^{N} \exp\left(\frac{\log \boldsymbol{x}_t^i(v') + \xi_{v'}}{\mathrm{T}_{\text{sample}}}\right)}, \tag{6}$$

where $\boldsymbol{x}_t^i(v)$ is the probability of token v in the vocabulary, ξ_v is drawn from the Gumbel$(0,1)^1$ distribution for token v, and $\mathrm{T}_{\text{sample}} > 0$ is the temperature parameter controlling the smoothness of the output. This enables gradient propagation during the projection step, while closely approximating the discrete $\arg\max$ operation.

Augmented Lagrangian projection. Consider a generic constraint on a sequence $\boldsymbol{x}$ defined via a measurable property given by a function $g_i(\boldsymbol{x})$. For instance, $g_i(\cdot) : \mathbb{R}^{L\times N} \to \mathbb{R}^+$ might evaluate molecular synthetic accessibility (see Section 5.1) or even evaluate molecular structural properties (see Section 5.2), or even serve as a black-box function computed by an external routine, as in our molecular novelty sequence generation application 5.3). To guide the projection operations, we require a measure of constraint violation that can later inform the parametrization of our projection update rule; for ease of presentation, we express the constraint in the form $g_i(\boldsymbol{x}) < \tau_i$, where $\tau_i \geq 0$ represents an acceptable threshold that must not be exceeded. To quantify by how much a given sequence violates the constraint, we define

$$\Delta g_i(\tilde{\phi}(\boldsymbol{x}_t)) = \max\left(0, g_i(\tilde{\phi}(\boldsymbol{x}_t)) - \tau_i\right),$$

where $g = (g_1, \ldots, g_m)$ can be treated as series of functions corresponding to a series of thresholds $\tau = (\tau_1, \ldots, \tau_m)$, and $\Delta g = (\Delta g_1, \ldots, \Delta g_m)$ is defined in analogously quantifying m constraints.

In practice, Δg is non-linear, and, thus, to implement Equation (5), we adopt an augmented Lagrangian approach [29]. In augmented Lagrangian methods, the problem constraints are incorporated into the objective of a minimizer via Lagrange multipliers λ and a quadratic penalty term μ. Let $\boldsymbol{x}_t'$ be the probability distribution after the denoising step at diffusion time t. We introduce a projected distribution $\boldsymbol{y}$, which is iteratively updated to reduce the constraint violations (measured by the score Δg) while remaining close (in KL-divergence) to $\boldsymbol{x}_t'$. Concretely the augmented Lagrangian dual function is defined as:

$$\mathcal{L}_{\text{ALM}}(\boldsymbol{y}, \boldsymbol{x}_t'; \lambda, \mu) = D_{\text{KL}}\left(\boldsymbol{x}_t' \| \boldsymbol{y}\right) + \sum_{i=1}^{m} \lambda_i \Delta g_i(\tilde{\phi}(\boldsymbol{y})) + \frac{\mu_i}{2} \Delta g_i(\tilde{\phi}(\boldsymbol{y}))^2$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is a non-negative Lagrange multiplier and $\mu = (\mu_1, \ldots, \mu_m)$ is a non-negative quadratic penalty term. When using a Lagrangian function, the primal optimization problem becomes

$$\arg\min_{\boldsymbol{y}} \mathcal{L}_{\text{ALM}}(\boldsymbol{y}, \boldsymbol{x}_t'; \lambda, \mu),$$

[3] We assume a greedy decoding scheme as is standard to current discrete diffusion generation models.

and is a lower bound of the original projection operator (5) by weak duality [29]. To obtain the strongest Lagrangian relaxation of the projection, the *Lagrangian dual* can be used to find the best Lagrangian multipliers λ and penalty terms μ, i.e.,

$$\arg\max_{\lambda,\mu}\Big(\arg\min_{\boldsymbol{y}}\big(\mathcal{L}_{\text{ALM}}(\boldsymbol{y},\boldsymbol{x}'_t;\lambda,\mu)\big)\Big). \tag{7}$$

In practice, the Lagrangian dual is a strong approximation of the original problem (our projection into the constraint space).

The optimization of (7) proceeds iteratively, following a gradient-based update on $\boldsymbol{y}$ while dynamically adjusting the Lagrange multiplier λ and penalty coefficient μ. Specifically, we perform the following updates [30]:

$$\boldsymbol{y} \leftarrow \boldsymbol{y} - \eta\nabla_{\boldsymbol{y}}\mathcal{L}_{\text{ALM}}(\boldsymbol{y},\boldsymbol{x}'_t;\lambda,\mu) \tag{8a}$$

$$\lambda \leftarrow \lambda + \mu\Delta g(\boldsymbol{y}^\star), \tag{8b}$$

$$\mu \leftarrow \min(\alpha\mu,\mu_{\max}), \tag{8c}$$

where η is the gradient step size, $\alpha > 1$ is a scaling factor that progressively increases μ over iterations, and $\mu_{\max}$ is an upper bound on the penalty term. These updates drive $\boldsymbol{y}$ as close as possible to satisfy $\Delta g(\tilde{\phi}(\boldsymbol{y}^\star)) \leq \tau$ while also ensuring it remains close to the original denoised distribution $\boldsymbol{x}'_t$. Pseudocode is provided for our implementation in Algorithm 1.

Algorithm 1: Augmented Lagrangian

Init: $\lambda, \mu, \eta, \alpha, \delta$;
Input: probability distribution $\boldsymbol{x}'_t$
$\boldsymbol{y} \leftarrow \boldsymbol{x}'_t$;
while $\Delta g(\boldsymbol{y}^*) < \delta$ **do**
  for $j \leftarrow 1$ **to** `max_inner_iter` **do**
    $\mathcal{L}_{KL} \leftarrow D_{\text{KL}}(\boldsymbol{x}'_t\|\boldsymbol{y})$
    $\mathcal{L}_{\text{viol}} \leftarrow \sum_{i=1}^{m} \lambda_i\Delta g_i(\tilde{\phi}(\boldsymbol{y})) + \frac{\mu_i}{2}\Delta g_i(\tilde{\phi}(\boldsymbol{y}))^2$
    $\mathcal{L}_{\text{ALM}} \leftarrow \mathcal{L}_{KL} + \mathcal{L}_{\text{viol}}$
    $\boldsymbol{y} \leftarrow \boldsymbol{y} - \eta\,\nabla_{\boldsymbol{y}}\mathcal{L}_{\text{ALM}}$
  $\lambda \leftarrow \lambda + \mu\,\Delta g(\boldsymbol{y}^*)$
  $\mu \leftarrow \min(\alpha\mu,\ \mu_{\text{max}})$
$\boldsymbol{x}_s \leftarrow \boldsymbol{y}$;
Output: $\boldsymbol{x}_s$

As shown in the next section, there is a high degree of flexibility in how these constraints can be implemented. For instance, they can be implemented as surrogate models (e.g., a classifier) that can be used to provide a continuous score, allowing for smooth gradient-base updates, as shown above.

Theoretical justification. The next result shows that the constrained reverse diffusion process converges to samples within the feasible region $\boldsymbol{C}$ while also keeping their distribution close to the data manifold, thus ensuring that the generated samples are both valid and realistic. Let $D_{\text{KL}}(\boldsymbol{x}_t,\boldsymbol{C}) = \inf_{\boldsymbol{y}\in\boldsymbol{C}} D_{\text{KL}}(\boldsymbol{y}\|\boldsymbol{x}_t)$ denote the KL divergence from $\boldsymbol{x}_t$ to the set $\boldsymbol{C}$.

Theorem 4.1 (Convergence of CDD)**.** *Let $\boldsymbol{C}$ be non-empty and β-prox-regular in the sense of [31, Def. 13.27], and the score network satisfy $\|\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t)\| \leq G$ (a standard consequence of the bounded-data domain after normalization). Then, for positive step sizes $\gamma_t, \leq \frac{1}{2G^2}\beta$, the following inequality holds for the distance to the feasible set $\boldsymbol{C}$:*

$$D_{\text{KL}}(\boldsymbol{x}'_s,\boldsymbol{C}) \;\leq\; (1-\boldsymbol{\alpha}_t)\,D_{\text{KL}}(\boldsymbol{x}'_t,\boldsymbol{C}) + \boldsymbol{\alpha}_{t+1}^2 G^2, \qquad \text{(non-asymptotic feasibility)}$$

where $\boldsymbol{\alpha}_t$ is proportional to the discrete Langevin step size γ_t and G bounds the score norm.

Theorem 4.1 shows that the distance to the feasible set $\boldsymbol{C}$ decreases at a rate of $1-\boldsymbol{\alpha}_t$ at each step (up to an additive $\boldsymbol{\alpha}_t^2 G^2$ noise). This implies ε-feasibility after $\mathcal{O}(\boldsymbol{\alpha}_{\min}^{-1})$ steps, with $\boldsymbol{\alpha}_{\min} \propto \min_t \gamma_t$, and a cumulative KL drift within $\mathcal{O}(\sum_t \boldsymbol{\alpha}_t)$. The theorem assumes β-prox-regularity, which provides a relaxation of typical convexity assumptions, and implies that for each viable point and normal direction, small perturbations still project uniquely and smoothly back to $\mathbf{C}$. Theorem proofs are provided in Appendix B.

5 Experiments

To empirically demonstrate the advantage provided by CDD, we compare our method against similarly sized autoregressive and diffusion-based generative models. Specifically, we benchmark our performance on constrained sequence generation against discrete diffusion baselines MDLM and UDLM [7, 8] and a similarly sized standard transformer based auto regressive model. As demonstrated by [7] UDLM outperforms MDLM [8] for sequence generation tasks with smaller vocabulary sizes. Consequently, we use UDLM as the base discrete diffusion model and for each

Molecules (Synthetic Accessibility)							
Model	**Valid** [↑]	**Novel** [↑]	**QED** [↑]	**Viol (%)** $\tau=3.0$	$\tau=3.5$	$\tau=4.0$	$\tau=4.5$
AR	1023	0	0.46	91.6	78.4	62.3	42.0
$\text{AR}_{\text{FUDGE } \gamma=7}$	925	6	0.48	11.1	9.8	9.6	9.2
MDLM	596	20	0.45	85.9	73.7	61.1	44.0
$\text{MDLM}_{\text{D-CFG } \gamma=3}$	772	3	0.41	87.8	73.9	54.2	22.5
$\text{MDLM}_{\text{D-CBG } \gamma=3}$	436	14	0.37	50.5	48.6	46.1	44.7
UDLM	895	21	0.47	89.4	88.0	58.1	37.8
$\text{UDLM}_{\text{D-CFG } \gamma=5}$	850	18	0.47	80.6	58.6	35.9	13.9
$\text{UDLM}_{\text{D-CBG } \gamma=10}$	896	28	0.47	90.1	77.8	58.6	37.7
$\text{CDD}_{\tau=3.0}$ (Ours)	353	**36**	**0.63**	**0.0**	**0.0**	**0.0**	**0.0**
$\text{CDD}_{\tau=3.5}$ (Ours)	863	22	0.62	–	**0.0**	**0.0**	**0.0**
$\text{CDD}_{\tau=4.0}$ (Ours)	936	31	0.61	–	–	**0.0**	**0.0**
$\text{CDD}_{\tau=4.5}$ (Ours)	938	33	0.58	–	–	–	**0.0**

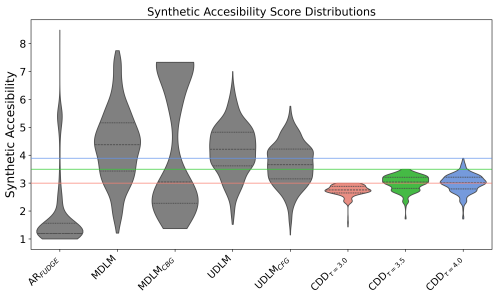

Figure 3: **Left:** Results for synthetic accessibility constrained molecule generation constraints. QED and constraint violations are reported for only valid molecules, and novel molecules must be valid and have no violation ($\tau \leq 3.0$). **Right:** Synthetic Accessibility score distributions for CDD versus competing baselines.

application, CDD uses configurations as described in [7] unless otherwise specified. Additional experiment details are available in Appendix A.

In this section we show the ability of CDD to impose constraints on a scientific task, specifically in the domain of molecular generation. In this experiment we generate SMILES strings [1], a linear representation of a molecule's structure with a vocabulary consisting of a limited vocabulary and a grammar dictating molecular validity. Recent advances in generative modeling have enabled the design of molecular sequences by leveraging techniques from natural language processing. Despite their impressive ability to optimize for specific chemical properties, these models often generate molecules that fall short of practical requirements, either by producing compounds that are difficult to synthesize or by closely mimicking existing structures. Furthermore, generated molecules often fail to maximize qualitative properties such as drug-likeness (QED) [32] that are critical to the practicality of the generations and used as a qualitative metric for the experiments discussed in this section.

We target three constraint applications:

1. **Synthetic accessibility**,which ensures that generated molecules can be synthesized in a laboratory setting [33].
2. **Toxicity**, which prohibits specific structural properties indicative of chemical toxicity or instability.
3. **Novelty**, which guarantees that the generated molecules are not already present in training datasets.

Across these regimes, CDD achieves *zero constraint violations* while maintaining high validity and QED, and substantially increases the fraction of novel molecules under equivalent guidance conditions. For each constraint, we implement a dedicated projection operator, described in A, that imposes hard limits on the generation process, resulting in final outputs that adhere precisely to our desired standards of synthetic accessibility or novelty.

5.1 Synthetic accessibility constraints.

Synthetic accessibility is commonly assessed using black-box, non-differentiable functions. To enable gradient-based optimization, we train a surrogate model on the QM9 dataset [34], where training labels are derived from RDKit's synthetic accessibility score [35]. This allows us to approximate a differentiable function $g(\cdot)$ for synthetic accessibility. While our molecular generation framework employs this surrogate model to optimize for synthetic accessibility during training, the actual assessment of accessibility violations is conducted by a separate, black-box external model. This external evaluation rigorously measures the degree to which generated molecules comply with the synthetic accessibility criteria, using a series of thresholds ($\tau = 3.0, 3.5, 4.0$, and 4.5).

For the discrete diffusion baselines, we adapt the guidance mechanisms provided by [8] for QED by retraining on synthetic accessibility labels for Classifier-Based Guidance (CBG) and Classifier-Free Guidance (CFG). A similar adaptation is applied for the autoregressive baselines employing FUDGE guidance [16]. The results, summarized in Figure 3 (left), confirm that CDD achieves perfect compliance across all thresholds. This 0% violation rate is achieved, all while generating a competitive number of valid molecules and exhibiting the highest drug-likeness scores.

Molecules (Novelty)			
Model	**Valid & Novel**	**QED**	**Viol (%)**
No Guidance			
AR	11	0.41	98.93
MDLM	271	<u>0.45</u>	<u>54.53</u>
UDLM	<u>345</u>	**0.46**	61.45
CDD (Ours)	**511**	<u>0.45</u>	**0.00**
CFG			
$\text{AR}_{\text{D-CFG } \gamma = 3}$ †	79	<u>0.60</u>	91.61
$\text{MDLM}_{\text{D-CFG } \gamma = 3}$ †	<u>96</u>	<u>0.60</u>	<u>69.82</u>
$\text{UDLM}_{\text{D-CFG } \gamma = 5}$ †	64	**0.62**	93.69
$\text{CDD}_{\text{D-CFG } \gamma = 5}$ (Ours)	**251**	<u>0.60</u>	**0.00**
CBG			
$\text{AR}_{\text{FUDGE } \gamma = 7}$ †	53	**0.61**	94.28
$\text{MDLM}_{\text{D-CBG } \gamma = 3}$ †	<u>117</u>	0.58	<u>72.08</u>
$\text{UDLM}_{\text{D-CBG } \gamma = 10}$ †	64	**0.61**	93.59
$\text{CDD}_{\text{D-CBG } \gamma = 10}$ (Ours)	**355**	0.59	**0.00**

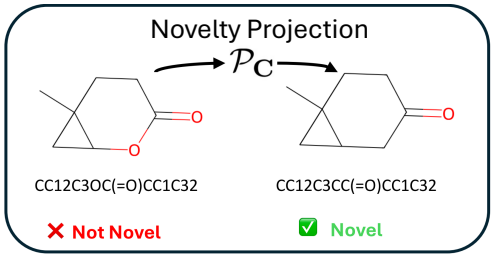

Figure 4: **Left:** Results for novelty projection with and without QED guidance. Violation represents percentage of valid, but not novel molecule generations. QED is reported for only novel molecules. Results denoted with † are as reported by Schiff et al. [8]. **Bold** and <u>underlined</u> values mark the best and second-best, respectively. **Right:** Projection illustration for a novelty application example.

5.2 Toxicity constraints

Model	Valid & Novel	QED	Three-membered heterocycles (Viol %)
AR	11	0.41	9.3
MDLM	271	0.45	22.2
UDLM	345	0.46	16.9
CDD	**351**	**0.47**	**0.0**

Table 1: Violation rates for three-membered heterocycles.

While the previous application instantiates CDD with surrogate, differentiable objectives here we show that CDD seamlessly extends to hard, non-differentiable structural rules specified by cheminformatics tooling, specifically substructure matching. For this application we prohibit the generation of molecules with three-membered heterocycles structures, as these are highly strained which makes them reactive and prone to ring opening. In medicinal chemistry this reactivity is linked to genotoxic liabilities and chemical instability and, contributing to off-target covalent binding [36]. At each reverse step, an external black box verifier [35] flags three-membered heterocycle substructures that are present in the molecules. If flagged, CDD solves a minimal-perturbation projection back to the feasible set, so constraints are enforced during sampling with empirical zero-violation guarantees, no retraining and no post-hoc rejection. This makes CDD attractive for safety-critical bio-generative settings: it is training-free and tool-agnostic as it works with black-box structure checkers, opening a line of applications where structural safety policies are enforced exactly rather than probabilistically. CDD attains *zero violations* on this constraint in addition to the highest Valid and Novel molecular generation amount as observed in Table 1.

5.3 Novelty constraints.

Novelty in molecule generation refers to the model's ability to produce new chemical structures that were not explicitly contained in its training set. In the context of generative modeling for drug design or other chemistry applications, novelty is often an important objective for the reasons of chemical space exploration and practical relevance. In this experiment, we generate SMILES strings constraining all valid generations to be novel. The novelty constraint is enforced at every denoising step. If the current candidate sequence already exists in an external database, a projection operator minimally perturbs its token-probability vector to yield an unseen molecule, thereby approximating a gradient step through the otherwise non-differentiable novelty indicator. Specifically, the novelty projection operator is a best-first traversal of the token-probability space: each token flip incurs a cost equal to its probability gap from the argmax, a priority queue retrieves the lowest-cost unseen sequence, and we then renormalize and cache it to prevent duplication. As it selects the sequence with the minimal cumulative flip cost, our distance metric over sequences, the procedure is mathematically equivalent to projecting the distribution onto the set of novel sequences. Repeating this operation throughout the diffusion trajectory guarantees that only new compounds are emitted while preserving the exploratory capacity of the discrete diffusion model.

We compare against baselines with no QED guidance alongside CFG and CBG QED guidance [8] to evaluate the impact of our novelty constraint. The results are shown in the right side of Figure 4. The CBG setting yields a 203.4% increase in novel molecule generation, while the CFG setting shows a 161.4% increase. Even without guidance, the method still produces 48.1% more novel molecules, with only a minimal reduction in the QED score. *These results are significant because they demonstrate that CDD can effectively generate novel molecules while maintaining a high QED score, which is crucial for many scientific tasks.*

6 Conclusion

In this paper we presented Constrained Discrete Diffusion (CDD), a novel method for imposing both structured and unstructured constraints during sequence generation. CDD relies on the adoption of discrete diffusion models which enable the consideration of entire sequences when imposing constraints during the generation process. Offering a key departure from current paradigms for controllable sequence generation, CDD integrates differentiable optimization methods into the diffusion reverse process. This integration, governed by a rigorous mathematical optimization framework, provides precise control for discrete sequence generation. The method is empirically validated on (i) molecular property adherence, (ii) toxic structural prohibition, and (iii) novelty, reporting state-of-the-art results for controllable generation in both terms of constraint adherence and quality metrics.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No 2234693. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This work was partially supported by NSF grants CAREER-2401285, RI-2533631, RI-2334936, DARPA under Contract No. #HR0011252E005, and UVA's National Security Data & Policy Institute, ODNI Contracting Activity #2024-24070100001 The work has also been supported by LLNL under Contract DE-AC52- 07NA27344 and by the LLNL-LDRD Program under Project No. 24-ERD-010 and 24-SI-008. This manuscript has been co-authored by Lawrence Livermore National Security, LLC under Contract No. DE-AC52-07NA27344 with the U.S. Department of Energy. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, DOE, DARPA, or DOD.

References

[1] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28 (1):31–36, 1988.

[2] Alain C. Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H. Nair, Philippe Schwaller, and Teodoro Laino. Automated extraction of chemical synthesis actions from experimental procedures. *Nature Communications*, 11(1):3601, 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-17266-6. URL `https://doi.org/10.1038/s41467-020-17266-6`.

[3] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, January 2018. ISSN 2374-7951. doi: 10.1021/acscentsci.7b00572. URL `http://dx.doi.org/10.1021/acscentsci.7b00572`.

[4] Nate Gruver, Samuel Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion, 2023. URL `https://arxiv.org/abs/2305.20009`.

[5] Wenhao Gao and Connor W. Coley. The synthesizability of molecules proposed by generative models, 2020. URL `https://arxiv.org/abs/2002.07007`.

[6] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023. URL `https://arxiv.org/abs/2107.03006`.

[7] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

[8] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

[9] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024.

[10] Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024.

[11] 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. *arXiv preprint arXiv:2409.02908*, 2024.

[12] Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured NLP tasks without finetuning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.674. URL `https://aclanthology.org/2023.emnlp-main.674`.

[13] Saibo Geng, Berkay Döner, Chris Wendler, Martin Josifoski, and Robert West. Sketch-guided constrained decoding for boosting blackbox large language models without logit access. *arXiv preprint arXiv:2401.09967*, 2024.

[14] Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *arXiv preprint arXiv:2010.12884*, 2020.

[15] Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*, 2021.

[16] Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.276. URL `http://dx.doi.org/10.18653/v1/2021.naacl-main.276`.

[17] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2019. URL `https://arxiv.org/abs/1802.04364`.

[18] Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. Gradient-based constrained sampling from language models. *arXiv preprint arXiv:2205.12558*, 2022.

[19] Afra Amini, Li Du, and Ryan Cotterell. Structured voronoi sampling. *Advances in Neural Information Processing Systems*, 36:31689–31716, 2023.

[20] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[21] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

[22] Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models, 2024.

[23] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

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

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

[26] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35:34532–34545, 2022.

[27] Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of langevin dynamics based algorithms for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[28] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[29] Stephen Boyd. Convex optimization. *Cambridge UP*, 2004.

[30] Ferdinando Fioretto, Pascal Van Hentenryck, Terrence W. K. Mak, Cuong Tran, Federico Baldo, and Michele Lombardi. Lagrangian duality for constrained deep learning. In *European Conference on Machine Learning*, volume 12461 of *Lecture Notes in Computer Science*, pages 118–135. Springer, 2020. doi: 10.1007/978-3-030-67670-4_8. URL `https://doi.org/10.1007/978-3-030-67670-4_8`.

[31] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[32] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[33] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.

[34] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

[35] A Patrícia Bento, Anne Hersey, Eloy Félix, Greg Landrum, Anna Gaulton, Francis Atkinson, Louisa J Bellis, Marleen De Veij, and Andrew R Leach. An open source chemical structure curation pipeline using rdkit. *Journal of Cheminformatics*, 12:1–16, 2020.

[36] Bidhan Chandra De, Wenjun Zhang, Chunfang Yang, Attila Mándi, Chunshuai Huang, Liping Zhang, Wei Liu, Mark W. Ruszczycky, Yiguang Zhu, Ming Ma, Ghader Bashiri, Tibor Kurtán, Hung-wen Liu, and Changsheng Zhang. Flavin-enabled reductive and oxidative epoxide ring opening reactions. *Nature Communications*, 13:4896, August 2022. doi: 10.1038/s41467-022-32641-1.

A Experimental Details

A.1 Molecular Generation

Training Dataset. For our molecule generation experiments, we utilize the QM9 dataset [34], as used by Schiff et al. [8]. This dataset comprises approximately 133,885 small organic molecules, each encoded as a SMILES string [1], which compactly represents the molecular structure through a sequence of characters. The SMILES representation facilitates discrete modeling by enabling the treatment of molecular generation as a sequence prediction task, making it particularly amenable to discrete diffusion approaches. By leveraging QM9, we can rigorously evaluate the performance of our generative models on tasks that require both the preservation of chemical validity and the precise control of molecular properties, aligning with established protocols in the literature.

A.1.1 Synthetic Accessibility Constraints

Surrogate Model. We train our surrogate model on the QM9 dataset [34] but manually label the training data with with RDKIT's synthetic accessibility score [35]. We finetune GPT2 (124M) to act as this surrogate model and directly output a score $s \in [0, 10]$.

FUDGE Implementation. For this setting, we follow the FUDGE implementation provided by [8]. However, while their guidance is trained on QED scores, labeling samples from the QM9 dataset with these scores, we adapt it to label the training data with RDKIT's computation of the synthetic accessibility scores.

Diffusion Guidance Implementation. Similar to the implementation of FUDGE, we adapt the guidance mechanisms provided by [8] for QED by retraining on synthetic accessibility labels. Otherwise, our implementation mirrors their approach for QED guidance.

ALM Implementation. For the ALM projection, we initialize using the following hyperparameters: $\lambda_{\text{init}} = 0.0, \mu_{\text{init}} = 1.0, \mu_{\text{max}} = 1000, \text{outer_iter}_{\text{max}} = 1000, \text{inner_iter}_{\text{max}} = 100, \eta = 1.0$.

A.1.2 Toxicity Constraints

Toxicity-Structure Projection Operator. Three-membered heterocycles are highly strained and reactive and are associated with instability and adverse liabilities [36]. We detect this via RDKit substructure matching, which is also used for validity checks. When a candidate is flagged, a projection operator proposes the least-cost structural edit that removes the observed structure while staying close to the model's score distribution: (i) *ring expansion* (insert one atom to yield a 4-/5-membered ring); else (ii) *ring opening* (break one ring bond to linearize the fragment); else (iii) *excision and reconnection* (remove the ring and reconnect substituents). After each edit, the molecule is sanitized and re-validated.

A.1.3 Novelty Constraints

Novelty Projection Operator. In order to project the molecule sequences into a novel set, we apply a best-first search (BFS) at the probability distribution level. We begin by defining a flip cost for forcing a token to become the new argmax; this cost is the difference between the current top token's probability and the candidate token's probability, or zero if the candidate is already top-ranked. To find a novel sequence at minimal total flip cost, we start with an empty sequence and expand it position by position, accumulating flip costs in a priority queue. Once a full sequence is constructed, we decode it and check if it is absent from our existing dataset. If it is indeed novel, we finalize the sequence, shift the probability distribution so that each selected token becomes the definitive argmax, and insert the resulting sequence into the dataset to prevent re-generation. This procedure systematically maintains high-likelihood sequences while avoiding those already present, terminating for each sample as soon as it finds a suitable novel result before proceeding to the next sample.

FUDGE Implementation. For this application, we similarly follow the FUDGE implementation and configuration in [8] for QED guidance.

Diffusion Guidance Implementation. We use classifier based guidance and classifier free guidance for QED as implemented in [8].

B Missing Proofs

Proof of Theorem 4.1

Proof. We begin by proving this bound holds for projected diffusion methods operating in the image space:

$$D_{\mathrm{KL}}(\boldsymbol{x}'_s, \boldsymbol{C}) \;\leq\; (1-\boldsymbol{\alpha}_t)\, D_{\mathrm{KL}}(\boldsymbol{x}'_t, \boldsymbol{C}) + \boldsymbol{\alpha}_{t+1}^2 G^2, \tag{9}$$

For ease of notation, we will denote subsequent timestep in terms of an arbitrary t, such that $\boldsymbol{x}_{t-1} = \boldsymbol{x}_s$ and the subsequent timestep after that is denoted $\boldsymbol{x}_{t-2}$, etc.

Consider that at each iteration of the denoising process, projected diffusion methods can be split into two steps:

1. **Gradient Step:** $\boldsymbol{x}'_t = \boldsymbol{x}_t + \gamma_t \underbrace{\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)}_{s_t}$
2. **Projection Step:** $\boldsymbol{x}_{t-1} = \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_t)$

These steps are sequentially applied in the reverse process to sample from a constrained subdistribution.

$$\boldsymbol{x}_t \to \overbrace{\boldsymbol{x}_t + \gamma_t s_t}^{\boldsymbol{x}'_t} \to \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_t) = \boldsymbol{x}_{t-1} \to \overbrace{\boldsymbol{x}_{t-1} + \gamma_{t-1} s_{t-1}}^{\boldsymbol{x}'_{t-1}} \to \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_{t-1}) = \boldsymbol{x}_{t-2} \ldots$$

By construction, $\boldsymbol{x}_{t-1} = \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_t) \in \mathbf{C}$. Next, let us define the projection distance to $\mathbf{C}$ as:

$$f(\boldsymbol{x}) = D_{\mathrm{KL}}(\boldsymbol{x}, \mathbf{C}) = D_{\mathrm{KL}}\left(\boldsymbol{x} \| \mathcal{P}_{\mathbf{C}}(\boldsymbol{x})\right)$$

Since $\mathbf{C}$ is β-prox regular, by definition the following hold:

- f is differentiable outside $\mathbf{C}$ (in a neighborhood)
- $\nabla f(\boldsymbol{x}) = 2(\boldsymbol{x} - \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}))$
- ∇f is L-Lipshitz with $L = \frac{2}{\beta}$

The standard "descent lemma" (or smoothness inequality) for L-smooth functions applies:

Lemma B.1. *$\forall \boldsymbol{x}, \boldsymbol{y}$ in the neighborhood of $\mathbf{C}$:*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 = \boxed{f(\boldsymbol{x}) + 2\langle \boldsymbol{x} - \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{1}{\beta}\|\boldsymbol{y} - \boldsymbol{x}\|^2}$$

Applying this lemma, let us use $\boldsymbol{x} = \boldsymbol{x}'_{t-1}$ and $\boldsymbol{y} = \boldsymbol{x}'_t$. Noting that $\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_t) = \boldsymbol{x}_{t-1}$, we get:

$$D_{\mathrm{KL}}(\boldsymbol{x}'_t, \mathbf{C}) \leq \underbrace{D_{\mathrm{KL}}(\boldsymbol{x}'_{t-1}, \mathbf{C})}_{\text{Term A}} + \underbrace{2\,\langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, \boldsymbol{x}'_t - \boldsymbol{x}'_{t-1} \rangle}_{\text{Term B}} + \underbrace{\frac{1}{\beta}\|\boldsymbol{x}'_t - \boldsymbol{x}'_{t-1}\|^2}_{\text{Term C}} \tag{$\star$}$$

Decomposing Term B. First, consider that since the step size is decreasing $\gamma_t \geq \gamma_{t-1}$:

$$\begin{aligned} \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2} &\leq (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}) + \gamma_{t-1} s_{t-1} \\ &\leq (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}) + \gamma_t s_{t-1} \end{aligned}$$

By the same rationale,

$$\boldsymbol{x}'_t - \boldsymbol{x}'_{t-1} \leq (\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) + \gamma_t (s_t - s_{t-1}). \tag{Definition B.1}$$

Proof of non-expansiveness of the projection operator. Next, we prove the non-expansiveness of the projection operator:

$$\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\| \leq 2\,\gamma_{t+1} G^2 \tag{$\mathcal{L}^+$}$$

Given $\boldsymbol{x}_t = \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_{t+1})$ and $\boldsymbol{x}_{t-1} = \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_t)$,

$$\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\| = \|\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_{t+1}) - \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_t)\| \leq \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|$$

since projections onto closed prox-regular sets are L-Lipshitz.

Now:

$$\begin{aligned} \boldsymbol{x}'_{t+1} &= \boldsymbol{x}_{t+1} + \gamma_{t+1} s_{t+1}; \\ \boldsymbol{x}'_t &= \boldsymbol{x}_t + \gamma_t s_t; \\ \boldsymbol{x}'_{t+1} - \boldsymbol{x}'_t &= (\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) + (\gamma_{t+1} s_{t+1} - \gamma_t s_t). \end{aligned} \tag{Definition B.2}$$

Making the projection residual,

$$\boldsymbol{x}_{t+1} - \boldsymbol{x}_t = \boldsymbol{x}_{t+1} - \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}'_{t+1})$$

orthogonal to the target space at $\boldsymbol{x}_t$ (and any vector of the form $s_{t+1} - s_t$). Thus, since $\|s_t\| \leq G \;\; \forall t$:

$$\|\boldsymbol{x}'_{t+1} - \boldsymbol{x}'_t\|^2 = \|\gamma_{t+1} s_{t+1}\|^2 + \|\gamma_t s_t\| \leq (\gamma_{t+1}^2 + \gamma_t^2) G^2$$

Taking the square root:

$$\|\boldsymbol{x}'_{t+1} - \boldsymbol{x}'_t\| \leq \sqrt{\gamma_{t+1}^2 + \gamma_t^2}\, G$$

Since $\gamma_{t+1} \geq \gamma_t$:

$$\begin{aligned} \|\boldsymbol{x}'_{t+1} - \boldsymbol{x}'_t\| &\leq \sqrt{2}\gamma_{t+1} G \\ &< 2\gamma_{t+1} G \end{aligned}$$

Finally, by applying Definition (B.1), $\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\| \leq \|\boldsymbol{x}'_{t+1} - \boldsymbol{x}'_t\|$, and thus:

$$\boxed{\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\| \leq 2\gamma_{t+1} G}$$

□

Now, prox-regularity gives:

$$\begin{aligned} \langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, \boldsymbol{x}'_t - \boldsymbol{x}'_{t-1} \rangle &\leq \beta \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 \\ &\leq 4\beta \gamma_{t+1}^2 G^2 \end{aligned} \tag{Bound B.1}$$

where the Bound B.1 is derived by applying ($\mathcal{L}^+$).

Since $\mathbf{C}$ in β-prox regular, for any point u near $\mathbf{C}$ and $v \in \mathbf{C}$:

$$\langle u - \mathcal{P}_{\mathbf{C}}(u), v - \mathcal{P}_{\mathbf{C}}(u) \rangle \leq \beta \|v - \mathcal{P}_{\mathbf{C}}(u)\|^2$$

Above, we substitute:

$$\begin{aligned} u &= \boldsymbol{x}'_t = \boldsymbol{x}_t + \gamma_t s_t \\ v &= \boldsymbol{x}_t \\ \mathcal{P}_{\mathbf{C}}(u) &= \boldsymbol{x}_{t-1} \end{aligned}$$

Now, expanding the inner product:

$$\begin{aligned}\langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, \boldsymbol{x}'_t - \boldsymbol{x}'_{t-1}\rangle &= \langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, (\boldsymbol{x}_t + \gamma_t s_t) - (\boldsymbol{x}_{t-1} + \gamma_{t-1} s_{t-1})\rangle \\ &\leq \langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, (\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) + \gamma_t(s_t - s_{t-1})\rangle \\ &\leq \langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, (\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\rangle + \langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, \gamma_t(s_t - s_{t-1})\rangle\end{aligned}$$

and since $\|s_t\| \leq G \quad \forall t$: $\langle s_{t+1}, s_t\rangle \leq \|s_{t+1}\|\|s_t\| \leq G^2$ so $\langle s_{t-1}, s_t\rangle - \|s_{t+1}\|^2 \leq G^2$, and:

$$\langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, \gamma_t(s_t - s_{t-1})\rangle \leq \gamma_t^2 G^2 \tag{Bound B.2}$$

By applying Definition (B.2):

$$\begin{aligned}\langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, \boldsymbol{x}'_t - \boldsymbol{x}'_{t-1}\rangle &= \langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, (\boldsymbol{x}_t - \boldsymbol{x}_t) + (\gamma_t s_t - \gamma_{t-1} s_{t-1})\rangle \\ &\leq \langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, (\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\rangle\end{aligned}$$

Therefore, by applying Bound (B.1) to the previous inequality and Bound (B.2) directly, Term B is upper bounded by:

$$\boxed{2\,\langle \boldsymbol{x}'_{t-1} - \boldsymbol{x}_{t-2}, \boldsymbol{x}'_t - \boldsymbol{x}'_{t-1}\rangle \leq 8\beta\gamma_{t+1}^2 G^2 + 2\gamma_t^2 G^2} \tag{Bound B.3}$$

Decomposing Term C. Next, we derive a bound on Term C in Eq. ($\star$). As already shown,

$$\|\boldsymbol{x}'_t - \boldsymbol{x}'_{t-1}\| \leq 4\gamma_t G,$$

given:

$$\begin{aligned}\boldsymbol{x}'_t - \boldsymbol{x}'_{t-1} &\leq \underbrace{(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})}_{\leq 2\gamma_{t+1}G} + \underbrace{\gamma_t(s_t - s_{t-1})}_{\leq 2G} \\ &\leq 4\gamma_{t+1} G\end{aligned}$$

Thus,

$$\boxed{\frac{1}{\beta}\|\boldsymbol{x}'_t - \boldsymbol{x}'_{t-1}\|^2 \leq \frac{16}{\beta}\gamma_{t+1}^2 G^2} \tag{Bound C.1}$$

Combining bounds (B.3) and (C.1) into ($\star$), and recalling that $\gamma_{t+1} \geq \gamma_t$:

$$D_{\mathrm{KL}}(\boldsymbol{x}'_t, \mathbf{C}) \leq \underbrace{D_{\mathrm{KL}}(\boldsymbol{x}'_{t-1}, \mathbf{C})}_{d} + \underbrace{(8\beta + 2 + \frac{16}{\beta})}_{K}\gamma_{t+1}^2 G^2$$

Now, we rewrite Term A, which for ease of notation we will refer to as d:

$$d = (1 - 2\beta\gamma_{t+1})d + 2\beta\gamma_{t+1} d$$

Thus:

$$\begin{aligned}D_{\mathrm{KL}}(\boldsymbol{x}'_t, \mathbf{C}) &\leq d - 2\beta\gamma_{t+1} d + 2\beta\gamma_{t+1} d + K\gamma_{t+1}^2 G^2 \\ &= (1 - 2\beta\gamma_{t+1})d + \left[2\beta\gamma_{t+1} d + K\gamma_{t+1}^2 G^2\right]\end{aligned}$$

Next, through Young's inequality, we simplify this expression further.

Theorem B.2. (Young's Inequality) $\forall u, v \geq 0, \epsilon > 0$*:*

$$uv \leq \frac{u^2}{2\epsilon} + \frac{\epsilon v^2}{2}$$

If we choose $u = \sqrt{2\beta\gamma_{t+1}}d$, $v = \sqrt{K}\gamma_{t+1}G$, and $\epsilon = \frac{2\beta d}{k\gamma_{t+1}G^2}$, then

$$\begin{aligned} uv &= \sqrt{2\beta\gamma_{t+1}}d \times \sqrt{K}\gamma_{t+1}G \\ &= \sqrt{2K}\gamma_{t+1}^{\frac{3}{4}}Gd \end{aligned}$$

Applying Young's Inequality:

$$\begin{aligned} uv &\leq \frac{u^2}{2\epsilon} + \frac{\epsilon v^2}{2} \\ &= \frac{2\beta\gamma_{t+1}d}{2(\frac{2\beta d}{K\gamma_{t+1}G^2})} + \frac{\epsilon v^2}{2} \\ &= \frac{K\gamma_{t+1}^2G^2}{2} + \frac{\epsilon v^2}{2} \\ &= \frac{K\gamma_{t+1}^2G^2}{2} + \left(\frac{1}{2} \times \frac{2\beta d}{K\gamma_{t+1}G^2} \times K\gamma_{t+1}^2G^2\right) \\ &= \frac{K\gamma_{t+1}^2G^2}{2} + \beta\gamma_{t+1}d \end{aligned}$$

Thus,

$$\sqrt{2K}\gamma_{t+1}^{\frac{3}{4}}Gd \leq \frac{K\gamma_{t+1}^2G^2}{2} + \beta\gamma_{t+1}d$$

Finally, taken altogether:

$$\begin{aligned} 2\beta\gamma_{t+1}d + K\gamma_{t+1}^2G^2 &\leq \beta\gamma_{t+1}d + \left(\frac{K\gamma_{t+1}^2G^2}{2} + \beta\gamma_{t+1}d\right) \\ &= 2\beta\gamma_{t+1}d + \frac{K}{2}\gamma_{t+1}^2G^2 \end{aligned}$$

Since $\gamma_{t+1} \leq \frac{\beta}{2G^2}$, then

$$\frac{K}{2}\gamma_{t+1}^2G^2 \leq \frac{1}{2}\left(8\beta + 2 + \frac{16}{\beta}\right)\frac{\beta^2}{4G^2} = \mathcal{O}(\beta^3)$$

which is bounded by $\gamma_{t+1}^2G^2$ for all $\beta \geq 0$.

Thus,

$$2\beta\gamma_{t+1}d + K\gamma_{t+1}^2G^2 \leq 2\beta\gamma_{t+1}d + \gamma_{t+1}^2G^2.$$

By substitution we obtain:

$$D_{\mathrm{KL}}(\boldsymbol{x}'_t, \mathbf{C}) \leq (1 - 2\beta\gamma_{t+1})D_{\mathrm{KL}}(\boldsymbol{x}'_{t-1}, \mathbf{C}) + \underbrace{\gamma_{t+1}^2G^2}_{\mathcal{O}(\beta^3)}$$

Finally, we reparameterize this, such that $\boldsymbol{\alpha}_t = 2\beta\gamma_t$, implying $\gamma_t = \frac{\boldsymbol{\alpha}_t}{2\beta}$. Plugging this in, we get:

$$\begin{aligned} D_{\mathrm{KL}}(\boldsymbol{x}'_t, \mathbf{C}) &\leq (1 - \boldsymbol{\alpha}_t)D_{\mathrm{KL}}(\boldsymbol{x}'_{t-1}, \mathbf{C}) + \gamma_{t+1}^2G^2 \\ &\leq (1 - \boldsymbol{\alpha}_t)D_{\mathrm{KL}}(\boldsymbol{x}'_{t-1}, \mathbf{C}) + \boldsymbol{\alpha}_{t+1}^2G^2 \end{aligned}$$

and thus,

$$\boxed{D_{\mathrm{KL}}(\boldsymbol{x}'_t, \mathbf{C}) \leq (1 - \boldsymbol{\alpha}_t)D_{\mathrm{KL}}(\boldsymbol{x}'_{t-1}, \mathbf{C}) + \boldsymbol{\alpha}_{t+1}^2G^2}$$

□