Neuro-Symbolic Generative Diffusion Models for Physically Grounded, Robust, and Safe Generation

Jacob K. Christopher CSK4SR@VIRGINIA.EDU
Michael Cardei NTR2RM@VIRGINIA.EDU
Jinhao Liang NJS4NU@VIRGINIA.EDU
Ferdinando Fioretto FIORETTO@VIRGINIA.EDU
Department of Computer Science, University of Virginia

Editors: G. Pappas, P. Ravikumar, S. A. Seshia

Abstract

Despite the remarkable generative capabilities of diffusion models, their integration into safety-critical or scientifically rigorous applications remains hindered by the need to ensure compliance with stringent physical, structural, and operational constraints. To address this challenge, this paper introduces *Neuro-Symbolic Diffusion* (NSD), a novel framework that interleaves diffusion steps with symbolic optimization, enabling the generation of certifiably consistent samples under user-defined functional and logic constraints. This key feature is provided for both standard and discrete diffusion models, enabling, for the first time, the generation of both continuous (e.g., images and trajectories) and discrete (e.g., molecular structures and natural language) outputs that comply with constraints. This ability is demonstrated on tasks spanning three key challenges: *(1) Safety*, in the context of non-toxic molecular generation and collision-free trajectory optimization; *(2) Data scarcity*, in domains such as drug discovery and materials engineering; and *(3) Out-of-domain generalization*, where enforcing symbolic constraints allows adaptation beyond the training distribution.

Keywords: Diffusion Models, Controllable Generation, Differentiable Optimization

1. Introduction

Diffusion models (Ho et al., 2020) are a class of generative AI models at the forefront of high-dimensional data creation and form the backbone of many state-of-the-art image and video generation systems (Rombach et al., 2022; Betker et al., 2023; Liu et al., 2024). This potential has also been recently extended to the context of discrete outputs, which is suitable for language modeling or combinatorial structure design, like chemical compounds or peptide design (Zheng et al., 2024; Lou et al., 2024; Shi et al., 2024). Diffusion models operate by progressively introducing controlled random noise into the original content and learning to reverse the process to reconstruct statistically plausible samples. This approach has shown transformative potential for engineering, automation, and scientific research through applications including generating trajectories for robotic agents in complex, high-dimensional environments or synthesizing new molecular structures with improved strength, thermal resistance, or energy efficiency (Carvalho et al., 2023; Watson et al., 2023).

However, as opposed to standard image synthesis tasks, scientific applications of diffusion models need to be controlled by precise mechanisms or properties that must be imposed on generations. *While diffusion models produce statistically plausible outputs, they are simultaneously unable to comply with fundamental physics, safety constraints, or user-imposed specifications (Motamed et al., 2025).* Violations of established principles and constraints not only undermine the utility

of generative models, but also erode their trustworthiness in high-stakes domains. For example, embodied agents powered by generative AI such as drones or robotic arms are susceptible to adversarial manipulations that bypass safety protocols (Robey et al., 2024). To date, these models have struggled to produce even simple trajectories that satisfy basic collision avoidance, let alone more stringent safety requirements in complex environments (Power et al., 2023). Similarly, in scientific and industrial applications such as autonomous bio-labs, systems may improperly model specifications or even react to adversarial triggers, potentially leading to synthesis of hazardous compounds (Wittmann, 2024). *Thus, there is a pressing need for generative models to satisfy physical, operational, and structural constraints that govern large-scale scientific and engineering challenges.*

Recently, Christopher et al. (2025) observed that a class of generative models can ground their induced distributions to a specific property. Inspired by this observation, this paper provides a step towards addressing the challenge of constraining generative models and developing a novel integration of symbolic optimization with generative diffusion models. The resulting framework, called *Neuro-Symbolic Diffusion* (NSD), enables the generation of outputs that are *certifiably consistent* with user-defined properties, ranging from continuous constraints, such as structural properties for material science applications or collision avoidance in motion-planning environments, to discrete constraints, including the prevention of toxic substructures in molecule generation tasks.

Contributions. The contributions of this study are as follows.

1. It develops a novel methodology *for integrating functional and logic constraints within generative diffusion models*. The core concept involves a tight integration of differentiable constraint optimization within the reverse steps of diffusion process and ensures that each generated sample respects user-imposed or domain-specific properties.
2. It shows that this approach is not only viable for generation within continuous subspaces but also effective in constraining token generation for discrete modalities, including domain-specific sequence generation for scientific discovery and open-ended language generation.
3. It provides theoretical grounding to demonstrate when and why constraint adherence can be certified during the neuro-symbolic generative process.
4. It presents an extensive evaluation across three key challenges: (1) *Safety*, demonstrated through non-toxic molecular generation and collision-free trajectory optimization; (2) *Data scarcity*, with applications in drug discovery and materials engineering; and (3) *Out-of-domain generalization*, where enforcing symbolic constraints enables adaptation beyond the training distribution.

These advances bring forward two key features that are important for the development of generative models for scientific applications: *Improved assurance*, e.g., the models can implement safety predicates needed in the domain of interest, such as natural language generation where prompts could be engineered to elicit harmful outputs, and *Improved generalization*, e.g., the imposition of knowledge and symbolic constraints dramatically improves the model generalizability across domains.

2. Preliminaries: Generative Diffusion Models

Diffusion models define a generative process by learning to reverse a *forward stochastic transformation* that progressively corrupts structured data into noise. The generative model then approximates the inverse of this transformation to restore the original structure, thereby allowing sampling from the learned distribution. The forward *noising* process $\{\boldsymbol{x}_t\}_{t=0}^{T}$ progressively corrupts data in a Markovian process, starting from $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0)$ and culminating into noise $\boldsymbol{x}_T \sim p(\boldsymbol{x}_T)$. Here, $p_{\text{data}}(\boldsymbol{x}_0)$ represents the distribution induced by real data samples, and $p(\boldsymbol{x}_T)$ is, by design, some

known distribution. The reverse process starts from $\boldsymbol{x}_T \sim p(\boldsymbol{x}_T)$ and produces samples $\boldsymbol{x}_0$ that follow p_{data}.

In this study, we consider two settings: (1) *continuous diffusion models* (Ho et al., 2020; Song et al., 2020) for data in $\mathbb{R}^d$ (e.g., images or trajectories), and (2) *discrete diffusion models* (Lou et al., 2024; Sahoo et al., 2024), which were recently introduced to handle discrete data (e.g., sequences of tokens representing natural language or molecular structures).

Diffusion models for continuous data. For data in $\mathbb{R}^d$, the forward diffusion process is often modeled as a *stochastic differential equation (SDE)* of the form $d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{B}(t)$, where $\boldsymbol{B}(t)$ denotes standard Brownian motion and $\beta(t)$ defines a noise schedule. As $t \to T$, the process asymptotically transforms data into an isotropic Gaussian distribution. The *reverse process*, which recovers the original data, follows a time-reversed SDE underlying *Langevin dynamics*:

$$d\boldsymbol{x}_t = \left[-\frac{1}{2}\beta(t)\boldsymbol{x}_t + \nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)\right]dt + \sqrt{\beta(t)}d\boldsymbol{B}(t). \tag{1}$$

However, since exact integration of this process is intractable, in practice, it is discretized into a *finite-step Markov chain*:

$$\boldsymbol{x}_{t-\Delta} = \boldsymbol{x}_t + \gamma_t \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \sqrt{2\gamma_t}\epsilon, \tag{2}$$

where $\boldsymbol{s}_\theta$ is a neural network that approximates the gradient of the log data distribution $\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)$, called the *score function*, and is used to guide the model toward high-density regions. Additionally, γ_t is the step size and ϵ is a Gaussian perturbation. In the deterministic limit (i.e., $\gamma_t \to 0$), this becomes a pure gradient ascent update on $\log p(\boldsymbol{x}_t)$.

Discrete diffusion models. For discrete data such as text tokens, each sample is a sequence $\boldsymbol{x}_0 = (\boldsymbol{x}_0^1, \ldots, \boldsymbol{x}_0^L)$ where each token $\boldsymbol{x}_0^i \in \mathbb{R}^V$ is represented as a one-hot vector over a vocabulary of size V. The forward process progressively corrupts the sequence by replacing tokens with noise, the marginal of which is defined as: $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \text{Cat}\left(\boldsymbol{x}_t;\ (1-\beta(t))\,\boldsymbol{x}_0 + \beta(t)\,\boldsymbol{\nu}\right)$, where $\beta(t) \in [0,1]$ is a schedule that increases with t, so that tokens are increasingly replaced by noise, and $\text{Cat}(\cdot;\boldsymbol{z})$ denotes a categorical distribution parameterized by probability vector $\boldsymbol{z} \in \Sigma^V$, where Σ^V denotes the V-dimensional simplex. Finally, $\boldsymbol{\nu}$ is a fixed categorical distribution, often concentrated on a special token, such as [MASK] (as in MDLM from Sahoo et al. (2024)) or chosen uniformly (as in UDLM from Schiff et al. (2024)). It models a process akin to that induced by the isotropic Gaussian in the continuous counterpart. As t increases, each token in $\boldsymbol{x}_t$ becomes less correlated with its original value, and approaches the noise distribution. The *reverse process* is represented as

$$\boldsymbol{x}_{t-\Delta} = \begin{cases} \text{Cat}\big(\boldsymbol{x}_{t-\Delta};\boldsymbol{x}_t\big), & \text{if } \boldsymbol{x}_t \neq \boldsymbol{\nu}, \\ \text{Cat}\Big(\boldsymbol{x}_{t-\Delta};\ \frac{\beta(t-\Delta)\,\boldsymbol{\nu} + (\beta(t)-\beta(t-\Delta))}{\beta(t)}\,\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\Big), & \text{if } \boldsymbol{x}_t = \boldsymbol{\nu}, \end{cases} \tag{3}$$

Since $\boldsymbol{x}_0$ is unknown at inference, it is approximated with $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$. Here, $\boldsymbol{x}_t$ represents a vector of *probability distributions* over tokens at each position in the sequence. The paper denotes with $\boldsymbol{x}_t^\star = \text{argmax}(\boldsymbol{x}_t)$ as the selected output sequence, where the argmax operator is applied independently to each member $\boldsymbol{x}_t^i$ of the sequence $\boldsymbol{x}_t$.

3. Related Work and Limitations

Despite their success, existing diffusion models struggle to enforce structured constraints. An approach developed to address this issue relies on sampling a conditional distribution $p_{\text{data}}(\boldsymbol{x}_0 \mid \mathbf{c})$, where $\mathbf{c}$ conditions the generation. This approach transforms the denoising process via

classifier-free guidance:

$$\hat{\boldsymbol{s}}_\theta \overset{\text{def}}{=} \lambda \times \boldsymbol{s}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) + (1-\lambda) \times \boldsymbol{s}_\theta(\boldsymbol{x}_t, t, \perp),$$

where $\lambda \in (0, 1)$ is the *guidance scale* and $\perp$ is a null vector representing non-conditioning (Ho and Salimans, 2022). These methods have demonstrated effectiveness in capturing physical design properties (Wang et al., 2023), positional awareness (Carvalho et al., 2023), and motion dynamics (Yuan et al., 2023). However, while conditioning can guide the generation process, it offers no reliability guarantees. This issue is illustrated in Figure 1 (red curve), which shows the magnitude of constraint violations observed in a physics-informed motion experiment simulating the dynamics of an object under a force-field influence (more details provided in Appendix E.4). Here, the conditional model fails to adhere to motion constraints as the diffusion steps evolve ($t \to 0$). Additionally, conditioning in diffusion models often necessitates the training of auxiliary classification or regression models, which requires additional labeled data. This is highly impractical in many scientific contexts of interest to this paper, where sample collection is expensive or extremely challenging.

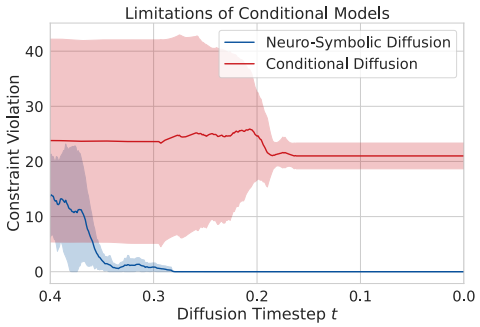


Figure 1: Conditional models fail to converge to feasible states while Neuro-Symbolic Diffusion produces no violations.

An alternative approach involves applying *post-processing steps* to correct deviations from desired constraints in the generated samples. This correction is typically implemented in the last noise removal stage (Giannone et al., 2023; Power et al., 2023; Mazé and Ahmed, 2023). However, this approach present two main limitations. First, the objective does not align with optimizing the diffusion model score function, and thus does not guide the model towards high-density regions. This inherently positions the diffusion model's role as ancillary, with the final synthesized data often resulting in a significant divergence from the learned (and original) data distributions. Second, these methods are reliant on a limited and problem specific class of objectives and constraints, such as specific trajectory "constraints" or shortest path objectives which can be integrated as a post-processing step (Giannone et al., 2023; Power et al., 2023).

To overcome these gaps and handle arbitrary symbolic constraints, our approach casts the reverse diffusion process as a differentiable constraint optimization problem, which is then solved through the application of repeated projection steps. The next section focuses on continuous models for clarity, but this reasoning extends naturally to discrete models, as shown in Appendix D.

4. Reverse Diffusion as Constrained Optimization

In traditional diffusion model sampling, the reverse process transitions a noisy sample $\boldsymbol{x}_T$ to $\boldsymbol{x}_0$ by reversing the stochastic differential equation in (1), which is discretized into an iterative Langevin dynamics update in (2). The key enabler for the integration of constraints in the diffusion process is the realization that *each reverse step can be framed as an optimization problem*. As shown in Xu et al. (2018), under appropriate regularity conditions, Langevin dynamics converges to a stationary distribution $p(\boldsymbol{x}_t)$, effectively maximizing $\log p(\boldsymbol{x}_t)$ (Christopher et al., 2025). As $t \to 0$, the variance schedule decreases and the noise term $\sqrt{2\gamma_t}\,\boldsymbol{\epsilon}$ vanishes, causing the update step to become deterministic gradient ascent on $\log p(\boldsymbol{x}_t)$. This perspective reveals that the reverse process can be viewed as *minimizing the negative log-likelihood of the data distribution*. The proposed method for constraining the generative process relies on this interpretation.

In traditional score-based models, at any point throughout the reverse process, $\boldsymbol{x}_t$ is unconstrained. When these samples are required to satisfy certain constraints, the objective remains unchanged, but the solution to this optimization must fall within a feasible region $\mathbf{C}$. Thus, the optimization problem formulation becomes:

$$\underset{\boldsymbol{x}_t \,:\, t\in(0,T]}{\text{Minimize}} \quad \int_{t=0}^{T} -\log p\big(\boldsymbol{x}_t \mid \boldsymbol{x}_0\big) \quad \text{subject to} \quad \boldsymbol{x}_t \in \mathbf{C}\,,\ \forall t \in (0,T]. \tag{4}$$

In practice, $\mathbf{C}$ is defined by the intersection of multiple (n) functional expressions or logic predicates: $\mathbf{C} \stackrel{\text{def}}{=} \bigwedge_{i=1}^{n} \phi_i(\boldsymbol{x})$, where each $\phi_i(\boldsymbol{x})$ is a predicate that returns 1 if $\boldsymbol{x}$ satisfies a condition and 0 otherwise. For example, these may be a series of properties that are required for a generated molecule to be a non-toxic chemical compound. The paper uses $\phi_{1:n}$ to denote the subset of n constraints in $\boldsymbol{C}$.

In discrete diffusion models, the score function can be similarly modeled through Concrete Score Matching as expounded on in Appendix D (Meng et al., 2022). Hence, the underlying optimization interpretation remains consistent: each denoising update seeks to move $\boldsymbol{x}_t$ closer to the high-density region of the learned distribution while respecting the constraints $\mathbf{C}$.

5. Neuro-Symbolic Generative Models

The score network $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ directly estimates the first-order derivatives of Equation (4) (excluding the constraints) and provides the necessary gradients for iterative updates defined in Equations (2) and (3). In the presence of constraints, however, an alternative iterative method is necessary to guarantee feasibility. This section illustrates how projected guidance can augment diffusion sampling and transform it into a constraint-aware optimization process. First, it formalizes the notion of a projection operator $\mathcal{P}_{\mathbf{C}}$, which finds the nearest feasible point to the input $\boldsymbol{x}$:

$$\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \underset{\boldsymbol{y}}{\operatorname{argmin}} \begin{cases} ||\boldsymbol{x}-\boldsymbol{y}||_2^2 & \text{s.t. } \boldsymbol{y} \in \mathbf{C}, \quad \text{if } \boldsymbol{x} \text{ is continuous,} \\ D_{\mathrm{KL}}\left(\boldsymbol{x} \parallel \boldsymbol{y}\right) & \text{s.t. } \boldsymbol{y}^\star = \operatorname{argmax}(\boldsymbol{y}) \in \mathbf{C}, \quad \text{if } \boldsymbol{x} \text{ is discrete.} \end{cases} \tag{5}$$

Because continuous diffusion operates in a multi-dimensional real space, $\boldsymbol{x} \in \mathbb{R}^d$, whereas discrete diffusion represents samples as $\boldsymbol{x} \in \Sigma^V$, the notion of proximity must be adapted accordingly. In continuous settings, Euclidean distance provides a natural measure of deviation from feasibility. At the same time, for discrete models, the underlying representations correspond to probability distributions; thus, the Kullback–Leibler (KL) divergence is chosen to quantify the minimal adjustment needed to satisfy the constraints. The optimization objective in Equation (5) defines a projection operator that minimizes a cost function, which we refer to as the *cost of projection*. In the continuous setting, this corresponds to the squared Euclidean distance, $\|\boldsymbol{y}-\boldsymbol{x}\|_2^2$, while in the discrete setting, it is determined by the KL divergence. More generally, we denote this projection cost as $D_{\text{cost}}(\boldsymbol{x}, \boldsymbol{y})$, representing the modality-specific distance minimized in the projection step.

To ensure that feasibility is maintained throughout the reverse process of the diffusion model, the sampling step is updated as:

$$\boldsymbol{x}_{t-\Delta} = \begin{cases} \mathcal{P}_{\mathbf{C}}\Big(\boldsymbol{x}_t + \gamma_t \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \sqrt{2\gamma_t}\epsilon\Big) & \text{if } \boldsymbol{x} \text{ is continuous,} \\ \mathcal{P}_{\mathbf{C}}\Big(\mathrm{Cat}\big(\cdot\,; \pi_\theta(\boldsymbol{x}_t, t)\big)\Big) & \text{if } \boldsymbol{x} \text{ is discrete.} \end{cases} \tag{6}$$

where $\gamma_t > 0$ is the step size, $\epsilon \sim \mathcal{N}(0, I)$, and $\pi_\theta(\boldsymbol{x}_t, t)$ is the predicted probability vector, as generalized from Equation (3). Hence, at each step of the Markov chain, a gradient update is applied to minimize the objective in Equation (4), while interleaved projections ensure feasibility throughout the sampling process. Importantly, convergence is guaranteed for convex constraint sets (see Section 6), and empirical results in Section 7 demonstrate the effectiveness of this approach, even in highly non-convex settings. Notably, the projection operators can be warm-started across iterations, providing a practical solution for efficiently handling regions with complex constraints.

Augmented Lagrangian Projection. To solve the projection subproblem $\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}_t)$ in each sampling step, the paper uses a Lagrangian dual method (Boyd, 2004), where the constraints are incorporated into a relaxed objective by using Lagrange multipliers λ and a quadratic penalty term μ. The augmented Lagrangian function is defined as

$$\mathcal{L}_{\text{ALM}}(\boldsymbol{y}, \lambda, \mu) = D_{\text{cost}}(\boldsymbol{x}_t,\, \boldsymbol{y}) + \lambda\, \tilde{\phi}(\boldsymbol{y}) + \tfrac{\mu}{2}\, \tilde{\phi}(\boldsymbol{y})^2,$$

where $\tilde{\phi}$ denotes a differentiable residual or constraint violation of the original (potentially non-differentiable) constraint function $\phi_{1:n}$. For example, consider a linear constraint $\phi = A\boldsymbol{y} \leq b$, then $\tilde{\phi} = \max(0, A\boldsymbol{y} - b)$. The iterative update follows a dual ascent strategy, where the variables $\boldsymbol{y}$ are optimized via gradient step on $\nabla_{\boldsymbol{y}} \mathcal{L}_{\text{ALM}}$, while the dual variables λ are updated by $\lambda \to \lambda + \mu\tilde{\phi}(\boldsymbol{y})$. Additionally, the penalty coefficients μ are increased adaptively by α to tighten constraint enforcement. This procedure continues until $\tilde{\phi}(\boldsymbol{y}) < \delta$ or the maximum iteration count is reached, returning a feasible $\boldsymbol{y}$ as $\boldsymbol{x}_{t-\Delta}$, as illustrated in Algorithm 1. Note that for convex constraint sets, the augmented Lagrangian method provides strong theoretical guarantees for exact convergence to the projection onto the feasible set (Boyd, 2004). Specifically, if Slater's condition holds (i.e., there exists a strictly feasible point), then the method guarantees convergence to the primal solution satisfying the constraints. This feature is key for several applications of interest to this work (see Section 7).

Algorithm 1 Augmented Lagrangian Projection

Input: $\boldsymbol{x}_t$, λ, μ, γ, α, δ
$\boldsymbol{y} \leftarrow \boldsymbol{x}_t$
while $\tilde{\phi}(\boldsymbol{y}) < \delta$ **do**
  for $j \leftarrow 1$ **to** *max_inner_iter* **do**
    $\mathcal{L}_{\text{ALM}} \leftarrow D_{\text{cost}}(\boldsymbol{x}_t,\, \boldsymbol{y}) + \lambda\, \tilde{\phi}(\boldsymbol{y}) + \frac{\mu}{2}\, \tilde{\phi}(\boldsymbol{y})^2$
    $\boldsymbol{y} \leftarrow \boldsymbol{y} - \gamma\, \nabla_{\boldsymbol{y}} \mathcal{L}_{\text{ALM}}$
  end
  $\lambda \leftarrow \lambda + \mu\, \tilde{\phi}(\boldsymbol{y});\;\; \mu \leftarrow \min(\alpha\mu,\, \mu_{\max})$
end
$\boldsymbol{x}_{t-\Delta} \leftarrow \boldsymbol{y}$
return $\boldsymbol{x}_{t-\Delta}$

Notice that, for discrete variables, the projection $\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}_t)$ must be imposed on the decoded sequence $\boldsymbol{y}^\star = \text{argmax}(\boldsymbol{y})$. Because the $argmax$ operator is not differentiable, this paper adopts a Gumbel-Softmax relaxation (Jang et al., 2016) to preserve gradient-based updates. Details on this relaxation are provided in Appendix C, and further technical aspects of the augmented Lagrangian scheme are discussed in Appendix B.

By incorporating constraints throughout the sampling process, the interim learned distributions are steered to comply with these specifications. The effectiveness of this approach is empirically evident from Figure 1 (blue curve): remarkably, as the reverse process unfolds, constraint violations steadily approach 0 and a theoretical justification for the validity of this approach is provided in the next section. A key distinction of this method, in contrast to prior approaches (Giannone et al., 2023; Power et al., 2023), is that it *optimizes the negative log-likelihood as the primary sampling objective*, maintaining consistency with standard unconstrained diffusion models while enforcing verifiable constraints. This provides a fundamental advantage: *it maximizes the probability of generating samples that conform to the data distribution while ensuring feasibility*. In contrast, existing baselines prioritize external constraints at the expense of distributional fidelity, often leading to significant deviations from the learned distribution, as shown in Section 7.

6. Effectiveness of Neuro-Symbolic Generation: A Theoretical Justification

This section focuses on two key outcomes of incorporating iterative projections during diffusion sampling: **(1)** As the sample $\boldsymbol{x}_t$ transitions toward the minimizer of the negative log-likelihood (the primary diffusion objective), each projection step needs only a small adjustment to maintain feasibility. Thus, the projected sampling remains closely aligned with the unconstrained score-based dynamics, causing minimal deviation from the main objective. **(2)** By keeping the sample near $\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}_t)$ throughout the sampling trajectory, any subsequent or "final" projection step becomes less costly (e.g., smaller Euclidean or KL distance).

Proofs for all theorems are provided in Appendix F and additional technical details in Appendix D. The analysis assumes a convex feasible region $\mathbf{C}$ and unifies results for both continuous (Euclidean) and discrete (KL-based) metrics (Christopher et al., 2025). Below, we detail the theoretical underpinnings using the update notation $\boldsymbol{x}_t \to \boldsymbol{x}_{t-\Delta}$ in place of traditional iterative indexing.
Consider an update step that transforms a sample $\boldsymbol{x}_t$ at diffusion time t into $\boldsymbol{x}_{t-\Delta}$ at time $t-\Delta$. Next, we use the update operator $\mathcal{U}_\theta(\boldsymbol{x}_t) \stackrel{\text{def}}{=}$ Eq. (1) if $\boldsymbol{x}_t$ is continuous or Eq. (3) if categorical.

We first establish a convergence criterion on the proximity to the optimum, showing that as diffusion progresses, the projected updates remain close to the highest-likelihood regions of the data distribution while respecting constraints.

Theorem 1 (Convergence Proximity) *If $\log p_{data}(\boldsymbol{x}_0)$ is convex, then there exists a minimum iteration $\bar{t}$ such that, for all $t \leq \bar{t}$, the following inequality holds: $\|\mathcal{U}_\theta(\boldsymbol{x}_t) - \Phi\|_2 \leq \|\rho - \Phi\|_2$, where ρ is the closest point to Φ, the global optimum of $\log p_{data}(\boldsymbol{x}_0)$, which can be reached via a single gradient step from any point in $\mathbf{C}$.*

Next, we show that incorporating projections systematically reduces the cost of enforcing feasibility, making the projection steps increasingly efficient as sampling progresses.

Theorem 2 (Error Reduction via Projection) *Let $\mathcal{P}_{\mathbf{C}}$ be the projection operator onto $\mathbf{C}$. For all $t \leq \bar{t}$, as defined by Theorem 1. Then,*

$$\mathbb{E}\left[\mathit{Error}(\mathcal{U}_\theta(\boldsymbol{x}_t), \mathbf{C})\right] \geq \mathbb{E}\left[\mathit{Error}(\mathcal{U}_\theta(\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}_t)), \mathbf{C})\right],$$

where $\mathit{Error}(\cdot, \mathbf{C})$ quantifies the cost of projection.

In essence, performing an update starting from a projected $\boldsymbol{x}_t$ yields a sample $\boldsymbol{x}_{t-\Delta}$ that is, in expectation, closer to the feasible set than an update without projection. A direct consequence is:

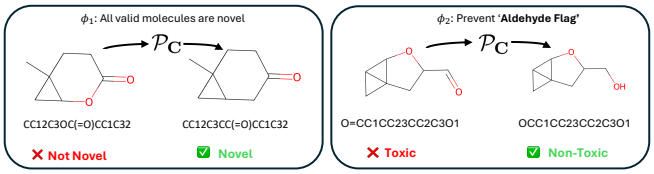
Corollary 3 (Convergence to Feasibility) *For any arbitrarily small $\xi > 0$, there exists a time t such that after the update*

$$\mathrm{Error}\big(\mathcal{U}_\theta(\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}_t)), \mathbf{C}\big) \;\leq\; \xi.$$

This result leverages the fact that the step size γ_t strictly decreases as t decreases, and thus, both the gradient magnitude and noise diminish. Consequently, the projection error approaches zero, implying that the updates steer the sample toward the feasible subdistribution of $p_{\text{data}}(\boldsymbol{x}_0)$.

Together, Theorem 2 and Corollary 3 explain why integrating the projection steps into the reverse update $\boldsymbol{x}_t \to \boldsymbol{x}_{t-\Delta}$ produces samples that conform closely to the imposed constraints.

Feasibility Guarantees. For an arbitrary density function, NSD provides feasibility guarantees for convex constraint sets. This assurance is critical in applications of interest to this paper, including the material design explored in Section 7.3 and physics-based simulations (Appendix E.4).



	Model	Novel	Novel & Non-Toxic	Viol (%) ϕ_1: novelty	Viol (%) $\phi_{2:6}$: non-toxic
Molecule Generation	AR	10.3 ± 2.3	5.3 ± 1.4	99.0 ± 0.2	40.2 ± 10.9
	MDLM	260.7 ± 16.4	108.0 ± 9.7	53.9 ± 3.1	35.3 ± 0.5
	UDLM	279.7 ± 22.7	132.3 ± 3.7	70.8 ± 2.4	38.1 ± 3.3
	NSD $_{\text{BRENK}}$	451.7 ± 19.5	392.0 ± 16.7	51.2 ± 2.0	$\mathbf{0.0 \pm 0.0}$
	NSD$_{\text{BRENK + Novel}}$	$\mathbf{533.3 \pm 8.7}$	$\mathbf{474.3 \pm 5.7}$	$\mathbf{1.4 \pm 0.3}$	$\mathbf{0.0 \pm 0.0}$

Figure 2: Results for Molecule Generation experiments constrained to be novel and non-toxic. On the left, we provide examples of the projection operators; on the right, the table outlines specific results.

7. Experiments and Evaluation

The evaluation of the symbolic diffusion approach focuses on three primary tasks designed to stress-test compliance with challenging constraints, with focus on *safety*, *data scarcity robustness*, and *out-of-domain generalization*. In all cases, we compare our method (NSD) against state-of-the-art baseline diffusion models and relevant ablations, as assessed by domain-specific qualitative metrics (i.e., path length for motion planning and FID scores for image generation) and frequency of constraint violations. Complimenting any domain specific baselines, the evaluation also includes, where applicable, a *conditional diffusion model (Cond)*, where constraints are applied as conditioning variables of the models and a *post-hoc correction (Post$^+$)* approach (projecting the final output only) to illustrate the importance of integrating constraints during sampling. We use identical neural network architectures and training procedures for the diffusion model across methods; thus, differences in performance can be attributed to constraint implementation rather than model capabilities. Due to space constraints, we fully elaborate all domain specifications in Appendix E. To demonstrate the broad applicability of NSD, the experimental settings showcase its capability in:

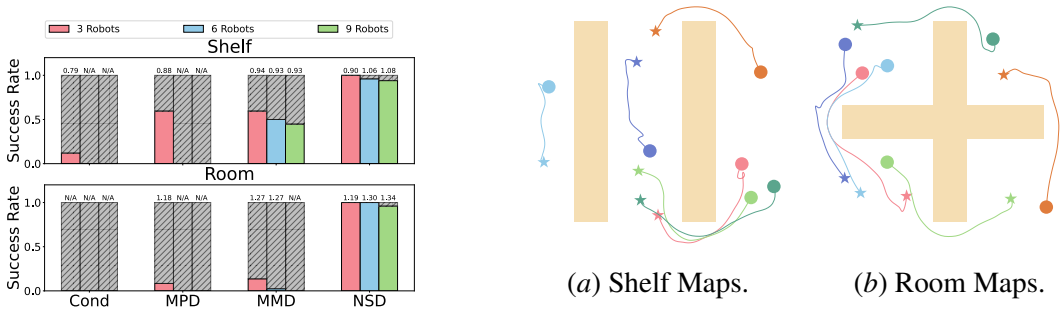
1. Enabling *safe, non-toxic* molecular generation and *out-of-domain* discovery (§7.1).
2. Handling *safety-critical* settings and *highly non-convex constraints* for motion planning (§7.2).
3. Facilitating microstructure design in *data scarce* settings for *out-of-domain* discovery (§7.3).

In addition, we test the ability of NSD to generate ODE-governed videos for *out-of-distribution* tasks (Appendix E.4), to prevent *harmful* text generation for natural language modeling (Appendix E.5), and to constrain supplementary *out-of-domain* molecule generation properties (Appendix E.1).

7.1. Molecule Generation for Drug Discovery (Safety and Domain Generalization)

In drug discovery, ensuring the *chemical safety and quality* of output molecules is critical. This experiment generates molecules in SMILES format (Weininger, 1988) using a uniform discrete diffusion model finetuned on the QM9 dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014). For this task, NSD is compared with MDLM (Sahoo et al., 2024) and UDLM (Schiff et al., 2024), the current state-of-the-art discrete diffusion models, and an autoregressive (AR) baseline with identical architecture and size to our diffusion model backbone.

The experiment enforces two key constraints: a novelty constraint (ϕ_1) that ensures generated molecules do not appear in the training set, and five BRENK substructure filters ($\phi_{2:6}$) that identify undesirable molecular fragments (e.g., aldehydes, three-membered heterocycles) linked to toxicity and the absence of drug-like characteristics. Critically, molecules flagged by BRENK often exhibit toxicity, reactivity, or other liabilities making them unsuitable for drug discovery (Brenk et al., 2008). Thus, for this study, we define 'non-toxic' molecules as those passing all five of the chosen BRENK filters (Appendix E.1). An illustration of the NSD correction mechanism employed to project the generated molecules is presented in Figure 2 (left).



(*a*) Shelf Maps. (*b*) Room Maps.

Figure 3: Evaluation on practical maps with three different numbers of robots. On the left, we assess failure rates (depicted by the gray regions of the bars) and average path length (values on top of the bars). On the right, we provide visualizations of the practical maps tested on.

Together, these constraints serve two purposes: **(1) Out-of-Distribution Generation:** the novelty constraint promotes *out-of-distribution generation*, which is essential for discovering new chemical compounds, and **(2) Safety-Critical Outputs:** the BRENK filters *ensure the sampled molecules are safe*, thereby improving their likelihood of success in downstream drug-development pipelines.

Figure 2 (right) reports the number of novel and non-toxic molecules generated, along with constraint violations (expressed as the percentage of generations that do not conform to the imposed requirements). While the diffuison baselines report substantial improvements with respect to the AR model, they frequently violate constraints. In contrast, NSD achieves *perfect adherence to safety constraints*, while also increasing the frequency of molecule generations that are novel, valid, and non-toxic *by over 3.5*$\times$, a remarkable improvement over the current state-of-the-art. Additionally, as detailed in Appendix E.1, we provide an evaluation of settings where NSD generations comply with strict thresholds on synthetic accessibility, the ease with which the generated molecules can be synthesized, further improving the practical utility.

7.2. Motion Planning for Autonomous Agents (Safety)

Next, we examine how NSD enforces collision avoidance in autonomous multi-agents settings using a *continuous* diffusion model. Two main challenges arise: **(1) Safety-Critical Outputs:** In real-world deployments, robots must avoid restricted or hazardous areas to ensure safe navigation in cluttered or dynamic environments. **(2) Highly Non-Convex Constraints:** Furthermore, the underlying problem is characterized by a *large number of non-convex* and temporal constraints, rendering the problem extremely challenging (more details in Appendix E.2).

We compare NSD to a conditional baseline and current state-of-the-art methods for multi-agent pathfinding: (1) Motion Planning Diffuion (MPD) originally designed for single-robot motion planning (Carvalho et al., 2023), and adapted here to multi-agent tasks for comparison and (2) Multi-robot Motion Planning (MMD), a recent method that integrates diffusion models with classical search techniques (Shaoul et al., 2024). Figure 3 (left) highlights the results on practical maps that feature multiple rooms connected by doors or constrained pathways for robot navigation. These scenarios require robots to not only avoid collisions (ϕ_1) but also coordinate globally to find feasible routes through shared spaces (ϕ_2). NSD sets a new *state-of-the-art* in feasibility and scalability. In Figure 3(*a*), MPD and MMD achieve 60% success with three robots but degrade significantly with more agents, with MMD dropping to 45% for nine robots. In contrast, NSD maintains high success rates: 100% for three robots, 96% for six, and 93% for nine. On room maps (Figure 3(*b*)), NSD

achieves perfect success rates for up to six robots and over 95% for nine robots, whereas MPD and MMD fail entirely as complexity increases. This trend persists across other environments (see Appendix E.2). This is remarkable, as NSD can enable effective coordination among multiple robots in shared, constrained spaces, ensuring collision-free, kinematically feasible trajectories.

7.3. Morphometric Property Specification (Data Scarcity and Domain Generalization)

Finally, this experiment focuses on a microstructure design task. Here, achieving specific morphometric properties is crucial for expediting the discovery of structure-property linkages. We consider an inverse-design problem in which a target porosity percentage, denoted by P(%), is desired. Here, the porosity is a measure of the percentage of 'damaged' pixels in the microstructure. This setting provides two particular challenges: **(1) Data Scarcity:** A key consideration in this context is the expense of generating training data, making data augmentation an important application of this problem. Obtaining real material microstructure images is expensive and time-consuming, with limited control over attributes such as porosity, crystal sizes, and volume fraction, often requiring a trial-and-error approach. Provided these costs, *our training data regime is very low*; we subsample a single 3,000× 3,000 pixel image to compose the dataset. **(2) Out-of-Distribution Constraints** Given the low amounts of data available, often the *desired porosity levels are unobserved in the training set*. Recall that NSD **guarantees** *strict adherence to specified porosity constraints*. Figure 4 illustrates the effectiveness of our method in generating microstructures with precise porosity levels as attempted by prior works employing conditional models (Chun et al., 2020). This demonstrates that our approach not only provides the highest fidelity to the training distribution but also outperforms baselines in producing valid microstructures as assessed by domain-specific heuristic metrics (Figure 6, bottom and Appendix E.3).

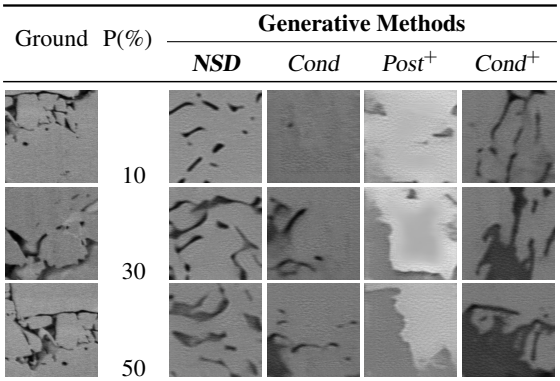


Figure 4: Samples and results from the morphometric property specification experiments.

8. Conclusion

This paper presented a novel framework that integrates symbolic optimization into the diffusion process, ensuring generative models can meet stringent physical, structural, or operational constraints. By enforcing these constraints continuously, rather than relying on post-processing approaches or soft guidance schemes, the proposed neuro-symbolic generative approach shows a unique ability to handle safety-critical tasks, cope with limited or skewed data, and generalize to settings beyond the original training distribution. Empirical evaluations across domains including toxic compound avoidance, motion planning, and inverse-design for material science illustrate this ability and provide a new state-of-the-art in utility and constraint adherence. As evidenced by this work, the capability to embed and process diverse symbolic knowledge and functional constraints into diffusion-based models paves the way for more trustworthy and reliable applications of generative AI in scientific, engineering, and industrial contexts.

Acknowledgments

This research is partially supported by NSF grant RI-2334936 and NSF CAREER Award 2401285. The authors acknowledge Research Computing at the University of Virginia for providing computational resources that have contributed to the results reported within this paper. The views and conclusions of this work are those of the authors only.

References

Kareem Ahmed, Eric Wang, Kai-Wei Chang, and Guy Van den Broeck. Neuro-symbolic entropy regularization. In *Uncertainty in Artificial Intelligence*, pages 43–53. PMLR, 2022.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Stephen Boyd. Convex optimization. *Cambridge UP*, 2004.

Ruth Brenk, Alessandro Schipani, Daniel James, Agata Krasowski, Ian Hugh Gilbert, Julie Frearson, and Paul Graham Wyatt. Lessons learnt from assembling screening libraries for drug discovery for neglected diseases. *ChemMedChem: Chemistry Enabling Drug Discovery*, 3(3):435–444, 2008.

Krysia B Broda, A d'Avila Garcez, and D Gabbay. Neural-symbolic learning system: foundations and applications, 2002.

Joao Carvalho, An T Le, Mark Baierl, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1916–1923. IEEE, 2023.

Jacob K Christopher, Stephen Baek, and Nando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2025.

Sehyun Chun, Sidhartha Roy, Yen Thi Nguyen, Joseph B Choi, HS Udaykumar, and Stephen S Baek. Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials. *Scientific reports*, 10(1):13307, 2020.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

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.

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.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Giorgio Giannone, Akash Srivastava, Ole Winther, and Faez Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation. *arXiv preprint arXiv:2305.18470*, 2023.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *arXiv preprint arXiv:2404.14743*, 2024.

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

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

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

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, NadineSchneider, Eisuke Kawashima, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, tadhurst cdd, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Vincent F. Scalfani, Rachel Walker, Kazuya Ujihara, Daniel Probst, Juuso Lehtivarjo, Hussein Faara, guillaume godin, Axel Pahl, and Jeremy Monat. rdkit/rdkit: 2024_09_5 (q3 2024) release, January 2025. URL https://doi.org/10.5281/zenodo.14779836.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models, 2024. URL https://arxiv.org/abs/2402.17177.

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.

François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Washington, DC*, 2023.

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.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models learn physical principles from watching videos?, 2025. URL https://arxiv.org/abs/2501.09038.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

Thomas Power, Rana Soltani-Zarrin, Soshi Iba, and Dmitry Berenson. Sampling constrained trajectories using composable diffusion models. In *IROS 2023 Workshop on Differentiable Probabilistic Robotics: Emerging Perspectives on Robot Learning*, 2023.

Raghunathan Ramakrishnan, Pavlo Dral, Matthias Rupp, and Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 08 2014. doi: 10.1038/sdata.2014.22.

Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J. Pappas. Jailbreaking llm-controlled robots, 2024. URL https://arxiv.org/abs/2410.13691.

R Tyrrell Rockafellar. Convergence of augmented lagrangian methods in extensions beyond nonlinear programming. *Mathematical Programming*, 199(1):375–420, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Lars Ruddigkeit, Ruud Deursen, Lorenz 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, 10 2012. doi: 10.1021/ci300415d.

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.

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

Yorai Shaoul, Itamar Mishani, Shivam Vats, Jiaoyang Li, and Maxim Likhachev. Multi-robot motion planning with diffusion models. *arXiv preprint arXiv:2410.03072*, 2024.

Iman Sharifi, Mustafa Yildirim, and Saber Fallah. Towards safe autonomous driving policies using a neuro-symbolic deep reinforcement learning approach. *arXiv preprint arXiv:2307.01316*, 2023.

Yifei Shen, Xinyang Jiang, Yifan Yang, Yezhen Wang, Dongqi Han, and Dongsheng Li. Understanding and improving training-free loss-based diffusion guidance. *Advances in Neural Information Processing Systems*, 37:108974–109002, 2025.

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.

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

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.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in Neural Information Processing Systems*, 35:23371–23385, 2022.

Tsun-Hsuan Wang, Juntian Zheng, Pingchuan Ma, Yilun Du, Byungchul Kim, Andrew Spielberg, Joshua Tenenbaum, Chuang Gan, and Daniela Rus. Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. *arXiv preprint arXiv:2311.17053*, 2023.

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.

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.

Maximilian Wittmann. Exploring the effect of anthropomorphic design on trust in industrial robots: Insights from a metaverse cobot experiment. In *2024 21st International Conference on Ubiquitous Robots (UR)*, pages 118–124. IEEE, 2024.

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.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Y Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. *Advances in Neural Information Processing Systems*, 37:22370–22417, 2025.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16010–16021, 2023.

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.

Appendix A. Extended Related Work

Neuro-symbolic frameworks. This paper's novel integration of symbolic constraints with generative models builds on foundational work in hybrid AI systems, blending the pattern recognition of neural networks with symbolic reasoning's structured constraints. Early approaches like cooperative architectures (Broda et al., 2002) established iterative feedback loops between neural and symbolic components, as seen in autonomous driving systems where visual detectors refine predictions via spatial logic rules (Sharifi et al., 2023). Parallel efforts in compiled architectures embedded symbolic operations directly into neural activations, enabling dynamic constraint enforcement in domains such as finance, where neurons encoded regulatory thresholds into credit scoring models (Ahmed et al., 2022).

Training-free correction. An alternative approach to enforcing desired properties in diffusion models is through training-free correction via gradient-based guidance. Inspired by methods such as Plug and Play Language Models (PPLM) (Dathathri et al., 2019), these techniques compute gradients from an external objective or constraint function at sampling time. Rather than relying on additional classifier training or extensive data labeling, the method directly adjusts the score estimates during the sampling process. Specifically, a loss function encoding the desired property is defined over the generated sample, and its gradient with respect to the sample is computed. Unlike model conditioning, which augments the score with a fixed conditioning signal, training-free correction dynamically refines the generation by continuously monitoring and correcting deviations from the target behavior (Guo et al., 2024; Shen et al., 2025). Such methods provide an alternative to existing conditioning approaches, but generally report worse performance than conditioning methods, due to inaccuracies in their gradients when the sample is at higher noise levels (Ye et al., 2025).

Appendix B. Augmented Lagrangian Method

Since $\tilde{\phi}$ is typically nonlinear and hard to enforce directly, we adopt an augmented Lagrangian approach (Boyd, 2004), which embeds the constraint $\tilde{\phi}(\boldsymbol{y}) \approx 0$ into a minimization objective with multipliers λ and a quadratic penalty μ. Let $\mathcal{U}_\theta(\boldsymbol{x}_t)$ be the sample after applying the denoising step at time t. We introduce a projected sample $\boldsymbol{y}$ that we iteratively refine to reduce violations of $\tilde{\phi}$ while remaining close to $\mathcal{U}_\theta(\boldsymbol{x}_t)$ under D_{cost}. The augmented Lagrangian is:

$$\mathcal{L}_{\text{ALM}}\big(\boldsymbol{y}, \lambda, \mu\big) = D_{\text{cost}}\big(\boldsymbol{x}_t, \boldsymbol{y}\big) + \lambda\, \tilde{\phi}(\boldsymbol{y}) + \tfrac{\mu}{2}\, \tilde{\phi}(\boldsymbol{y})^2.$$

Minimizing $\mathcal{L}_{\text{ALM}}$ yields a lower-bound approximation to the original projection. Its Lagrangian dual solves:

$$\arg\max_{\lambda,\mu}\Big(\arg\min_{\boldsymbol{y}} \mathcal{L}_{\text{ALM}}\big(\boldsymbol{y}, \lambda, \mu\big)\Big).$$

We optimize iteratively, updating $\boldsymbol{y}$ via gradient descent and adjusting λ and μ as follows Fioretto et al. (2020):

$$\boldsymbol{y} \leftarrow \boldsymbol{y} - \gamma \nabla_{\boldsymbol{y}} \mathcal{L}_{\text{ALM}}\big(\boldsymbol{y}, \lambda, \mu\big), \tag{7a}$$

$$\lambda \leftarrow \lambda + \mu\, \tilde{\phi}(\boldsymbol{y}), \tag{7b}$$

$$\mu \leftarrow \min(\alpha\mu, \mu_{\max}), \tag{7c}$$

where γ is the gradient step size, $\alpha > 1$ increases μ over iterations, and $\mu_{\max}$ is an upper bound. This drives $\boldsymbol{y}$ to satisfy $\tilde{\phi}(\boldsymbol{y}) \approx 0$ while staying close to $\mathcal{U}_\theta(\boldsymbol{x}_t)$. Noting that $\tilde{\phi}$ may be computed using a surrogate network, this optimization can be further grounded by directly using $\phi_{1:n}(\boldsymbol{y}) = 1$ as the termination condition; hence, assuming strong convergence properties (which are encouraged by the inclusion of the quadratic term (Rockafellar, 2023)), the projected sample will strictly satisfy the symbolic constraints as assessed by the reasoning test.

Appendix C. Discrete Sequence Relaxations

An important challenge to imposing gradient-based projections on discrete data sequences is providing a differentiable relaxation of our constraint satisfaction metric. This arises because we impose the constraint over the decoded version of the probability distributions, which is inherently discrete, making it not naturally differentiable. This poses a significant obstacle when one needs to backpropagate errors through operations that select discrete tokens or decisions. To overcome this limitation, we leverage a Gumbel-Softmax relaxation of the $\arg\max$ operator, denoted as ψ, which effectively bridges the gap between discrete and continuous representations.

More specifically, given a probability vector of token i, $\boldsymbol{x}_t^i$, where each component $\boldsymbol{x}_t^i(v)$ represents the probability assigned to token v from a vocabulary of size V, we approximate the hard, discrete decision of the $\arg\max$ function by constructing a continuous, differentiable approximation:

$$\psi(\boldsymbol{x}_t^i)(v) = \frac{\exp\left(\frac{\log \boldsymbol{x}_t^i(v) + g_v}{\mathrm{T}_{\text{sample}}}\right)}{\sum_{v'=1}^{V} \exp\left(\frac{\log \boldsymbol{x}_t^i(v') + g_{v'}}{\mathrm{T}_{\text{sample}}}\right)}.$$

Here, g_v is a random variable drawn independently from a Gumbel$(0, 1)$ distribution for each token v. The introduction of the Gumbel noise g_v perturbs the log-probabilities, thereby mimicking the stochasticity inherent in the discrete sampling process. The parameter $\mathrm{T}_{\text{sample}} > 0$ serves as a temperature parameter that governs the degree of smoothness of the resulting distribution. Lower temperatures make the approximation sharper and more similar to the original $\arg\max$ operator, while higher temperatures yield a smoother distribution that is more amenable to gradient-based optimization.

This relaxation not only facilitates the propagation of gradients through the projection step but also maintains a close approximation to the original discrete decision process. By incorporating the Gumbel-Softmax technique, we can integrate the $\arg\max$ operation into our model in a way that is compatible with gradient descent, ultimately enabling the end-to-end training of models that require discrete token decisions without sacrificing the benefits of differentiability (Jang et al., 2016).

Appendix D. Score Matching for Discrete Diffusion

Recall that in Section 2 we introduce the Euler discretized update step for Langevin dynamics:

$$\boldsymbol{x}_{t-\Delta} = \boldsymbol{x}_t + \gamma_t \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) + \sqrt{2\gamma_t}\epsilon$$

This directly allows us to formulate the objective of the reverse process from the given update rule, as shown in Equation (4). This representation of the diffusion sampling procedure is fundamental to our theoretical analysis. While our discussion focuses on continuous diffusion models, as noted earlier, the framework can be naturally extended to discrete diffusion models as well.

Particularly, we highlight that Langevin dynamics sampling algorithms used by continuous *score-based* diffusion models, whether applied directly (Song and Ermon, 2019) or through predictor-corrector frameworks (Song et al., 2020), can be utilized by discrete diffusion models. Notably, score-based discrete diffusion models leverage a discrete generalization of the score function, referred to as the *Concrete score* (Meng et al., 2022), to approximate the gradient of the probability density function $\log p_t(\boldsymbol{x}_t)$. As opposed to continuous score-based diffusion, where the gradient is directly applied to the representation of the sample (e.g., for image data the gradient will directly change pixel values), discrete models apply this gradient to the probability distributions which are sampled from to predict the final, discrete sequence. Despite this discrepancy, Concrete Score Matching provides an approach which mirrors 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.

As a final note, while many works do not explicitly adopt Concrete Score Matching as done by previous literature (Meng et al., 2022; Lou et al., 2024), the score function is often still implicitly modeled by the denoiser. For example, Sahoo et al. (2024) provide theoretical results demonstrating equivalence to a score-based modeling, supporting the extrapolation of our theoretical framework to models which employ simplified derivations of the Concrete Score Matching training objective.

Appendix E. Extended Experimental Results

E.1. Molecule Generation for Drug Discovery (Safety and Domain Generalization)

In this section further explain the setting for constrained molecular generation. We use UDLM (Schiff et al., 2024) as our underlying diffusion model architecture for NSD for this application. The task is to generate representations of molecule structures using SMILES sequences (Weininger, 1988), human readable strings that can be directly mapped to molecule compounds. We begin with an overview of the domain-specific benchmarks used in our evaluation. Then, we provide an extended version of Figure 2 (right), where we detail the violations for each specific BRENK test that is corrected by our projection operator. We then discuss the violations, their corresponding symbolics tests, and projection operators in detail. Finally, we introduce and explain the setting and results for constraining the synthetic accessibility of the molecules generated.

Additional benchmarks. To supplement our evaluation, we compare to several domain specific approaches:

1. **Autoregressive LLM (AR)**: An autoregressive transformer-based model, trained for molecule generations and sized to be comparable with the other diffusion-based architectures (100M parameters).
2. **Conditional Masked Diffusion Model (MDLM)**: A conditional masked discrete diffusion model implementation from Schiff et al. (2024) with guidance schemes in the subscript if applicable.
3. **Conditional Uniform Diffusion Model (UDLM)**: A conditional uniform discrete diffusion model implementation from Schiff et al. (2024) with guidance schemes in the subscript if applicable.

Symbolic test. For the purpose of generating safer, higher quality, and novel molecules we implement a total of six symbolic tests each corresponding with its own correction; in practice, our projection operator composes these corrections to find the nearest feasible point *at the intersection* of these constraints, $\boldsymbol{x} \in \mathbf{C}$.

Molecule Generation	Model	Novel & Non-Toxic	Viol (%)					
			ϕ_1: novelty	ϕ_2: Aldehydes	ϕ_3: Three-membered heterocycles	ϕ_4: Imines	ϕ_5: Triple bonds	ϕ_6: Isolated alkene
	AR	5.3 ± 1.4	99.0 ± 0.2	20.2 ± 6.2	9.3 ± 7.6	21.2 ± 13.9	10.9 ± 2.6	2.6 ± 4.1
	MDLM	108.0 ± 9.7	53.9 ± 3.1	9.5 ± 1.4	22.2 ± 2.1	11.0 ± 1.9	9.0 ± 1.5	10.3 ± 0.9
	UDLM	132.3 ± 3.7	70.8 ± 2.4	11.3 ± 1.4	16.9 ± 2.1	10.9 ± 2.6	8.0 ± 2.9	11.1 ± 1.6
	NSD BRENK	392.0 ± 16.7	51.2 ± 2.0	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
	NSD$_{\text{Novel + BRENK}}$	474.3 ± 5.7	1.4 ± 0.3	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$

Table 1: Extended results from Figure 2 (right).

1. **Novelty** ϕ_1: As defined in (Schiff et al., 2024), a generate molecule is considered valid if it can be parsed by the RDKit library (Landrum et al., 2025), and a molecule is novel if its valid, unique, and not present in the QM9 dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014). The test function ϕ_1 determines whether the current discretized molecular representation $\boldsymbol{x}^\star$ is within the training set $\mathcal{D}$, thus $\phi_1 \stackrel{\text{def}}{=} \boldsymbol{x}^\star \notin \mathcal{D}$. If ϕ_1 is not satisfied, the corresponding projection operation uses a best-first search to systematically flip tokens in a sequence, based on minimal probability "flip costs" to generate new sequences not yet in the dataset. Once a novel sequence is found, it is finalized, added to the dataset, and the model's token probabilities are adjusted to maintain high likelihood for the newly generated sequence. Specifically, we seek a novel sequence $\boldsymbol{x}^\star \notin \mathcal{D}$ by minimally altering the top-ranked tokens and flip cost denoted as D_{cost}. We find

$$\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{y}^\star \notin \mathcal{D}} D_{\text{cost}}(\boldsymbol{y}, \boldsymbol{x}),$$

   via a best-first search. Once a novel sequence is found, it is added to $\mathcal{D}$, and the distribution is updated so that $\boldsymbol{x}$ becomes the new top-ranked path, avoiding duplicates in future generations.
2. **Substructure Violations** $\phi_{2:6}$: In order to generate safer and less toxic molecules we use the blackbox BRENK filter (Brenk et al., 2008) provided by RDKit (Landrum et al., 2025) which offers various violation alerts which lead to a BRENK flag. While there are many of these potential substructure violations, we cover the five most frequent. For these violation we use RDKit to identify and flag these substructures. Thus, we can define $B = \{\, \boldsymbol{x}^\star \mid \text{BRENK}_{\text{Flag}}(\boldsymbol{x}^\star) = \text{True}\}$, where B is the set of molecules that the BRENK filter flags. Now, we can define the tests as: $\phi_{2:6} \stackrel{\text{def}}{=} \boldsymbol{x}^\star \notin B$ with each specific ϕ_i described below.
   (a) **Aldehydes** ϕ_2: Aldehydes feature a carbonyl group (`C=O`) in which the carbon is also bonded to at least one hydrogen (i.e., `R--CHO`). In SMILES notation, this typically appears as `C=O` where the carbon atom carries a hydrogen. In drug discovery, aldehydes are considered undesirable owing to their high reactivity and potential toxicity.

   To address flagged aldehydes, our method proceeds as follows:

      i. *Transform*: Identify the aldehyde (`C=O` with a hydrogen on the carbon) and attempt to convert it into either an alcohol (`R--CH2--OH`) or a methyl ketone (`R--C(=O)CH3`).
      ii. *Fallback*: If neither transformation produces a valid molecule, directly modify the carbonyl bond (e.g., force `C=O` to `C--OH`) or remove the oxygen entirely, thus eliminating the problematic functionality.

   These operations yield a molecular sequence that no longer violates the aldehyde-related BRENK filter.

   (b) **Three-membered heterocycles** ϕ_3: These are small ring systems composed of three atoms, at least one of which is a heteroatom (e.g., nitrogen, oxygen, etc.). Molecules containing

such rings are considered undesirable due to their high ring strain, reactivity, and potential toxicity. Typically, they are detected by filters that look for three-member rings containing a heteroatom.

To correct a flagged three-membered heterocycle, we use a stepwise approach:

i. *Ring expansion*: Insert a new carbon atom into one of the ring bonds, creating a larger, less-strained ring.

ii. *Bond breaking*: If expansion fails to produce a valid, non-flagged molecule, break one of the ring bonds to open the ring.

iii. *Complete removal*: If neither of the previous steps works, remove all bonds in the original three-membered ring entirely.

After each step, we check for validation and against the BRENK filter. The result is a structure that no longer violates the "three-membered heterocycle" constraint.

(c) **Imines** ϕ_4: An imine is a functional group containing a carbon–nitrogen double bond (C=N). These groups are often flagged due to potential instability and reactivity.

The corresponding operation employs a two-stage procedure:

i. *Initial fix*: Convert the double bond into a single bond and add a hydrogen atom to the nitrogen.

ii. *Fallback fix*: If the first approach fails or yields an invalid structure, remove or break the C=N bond entirely so that no imine remains.

(d) **Triple bonds** ϕ_5: Molecules containing triple bonds (denoted by # in SMILES) can be flagged due to concerns about reactivity, metabolic stability, or synthetic difficulty. To address such cases, we apply a simple transformation that replaces the triple bond character (#) with either a double bond (=) or a single bond (−), thus reducing the likelihood of reactivity or instability.

(e) **Isolated alkene** ϕ_6: Alkenes, represented by (=) can be flagged if they appear in undesired or isolated positions that may lead to reactivity or instability issues. To address this, when flagged, our method replaces the double bond character (=) with a single bond (−), effectively saturating the alkene. This ensures that the final structure does not violate the isolated-alkene BRENK constraint.

Next, to supplement the evaluations provided in the main text, we provide an additional setting where we constrain synthetic accessibility (SA) of the generated molecules below a strict threshold. For this setting, we consider that many of the generated molecules, while potentially novel and valid generations, cannot be directly synthesized due to the complexity of these compounds (Ertl and Schuffenhauer, 2009). Hence, we impose a constraint on the permissible synthetic accessibility of these outputs and compare to a series of conditional models (Table 2). Notably, our model yields a 0% violation rate, despite generating a competitive number of valid molecules and exhibiting the highest drug-likeness scores (referred to in the table as QED). These results demonstrate how the inclusion of constraint projection operators ensure that generated molecules not only scores well in property optimization but also adhere to synthetic feasibility requirements as determined by an independent, external standard.

	Model	Valid [↑]	Novel [↑]	QED [↑]	Viol (%) $\tau = 3.0$	$\tau = 3.5$	$\tau = 4.0$	$\tau = 4.5$
Molecules (Synthetic Accessibility)	AR	1023	11	0.46	91.6	78.4	62.3	42.0
	$\text{AR}_{\text{FUDGE } \gamma=7}$	925	13	0.48	11.1	9.8	9.6	9.2
	MDLM	596	271	0.45	85.9	73.7	61.1	44.0
	$\text{MDLM}_{\text{(D-CFG: } \gamma = 3)}$	772	53	0.41	87.8	73.9	54.2	22.5
	$\text{MDLM}_{\text{(D-CBG: } \gamma = 3)}$	436	21	0.37	50.5	48.6	46.1	44.7
	UDLM	895	<u>345</u>	0.47	89.4	88.0	58.1	37.8
	$\text{UDLM}_{\text{(D-CFG: } \gamma = 5)}$	850	69	0.47	80.6	58.6	35.9	13.9
	$\text{UDLM}_{\text{(D-CBG: } \gamma = 10)}$	896	**374**	0.47	90.1	77.8	58.6	37.7
	$\textbf{NSD}_{\tau=3.0}$	353	36	**0.63**	**0.0**	**0.0**	**0.0**	**0.0**
	$\textbf{NSD}_{\tau=3.5}$	863	91	<u>0.62</u>	-	**0.0**	**0.0**	**0.0**
	$\textbf{NSD}_{\tau=4.0}$	936	108	0.61	-	-	**0.0**	**0.0**
	$\textbf{NSD}_{\tau=4.5}$	938	121	0.58	-	-	-	**0.0**

Table 2: Molecule generation constrained to strict synthetic accessibility thresholds.

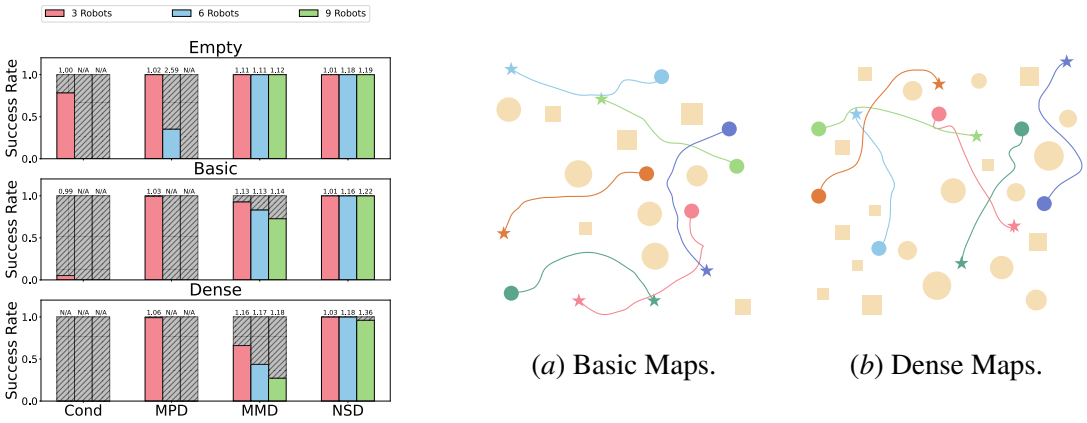
E.2. Motion Planning for Autonomous Agents (Safety)

For this task, we begin by assigning start and goal states for a series of agents in each respective environment. For practical maps (Figure 3), these positions fall in predefined zones that reflect real-world constraints, such as designated pickup and drop-off locations in a warehouse. For random maps (Figure 5), we assign these start and goal state randomly, constraining these to be collision-free locations, ensuring feasible solutions exist. To further ensure this, we discretize the environments and apply multi-agent pathfinding algorithms to verify the existence of collision-free solutions. If no valid assignment can be found, we regenerate the configuration.

Our results are evaluated by success rate (bars included in the figures), the frequency at which state-of-the-art methods find feasible solutions, and path length (at the top of the bars), a metric for the optimality of the solutions. Experiments are conducted with three, six, and nine robots, generating ten test cases per configuration. In addition to evaluation on practical map environments illustrated in Figure 3, we provide additional evaluation on random maps in Figure 5. Again, we see that NSD dramatically outperforms the baselines in its ability to generate feasible motion trajectories. This is particularly exasperated as we scale the number of agents and obstacles. Of the baselines, only MMD is able to ever provide feasible solutions for nine agents, although we note that it has much more frequent constraint violations than NSD on non-empty maps.

Additional benchmarks. To supplement our evaluation, we compare to several domain specific approaches:

1. **Conditional Diffusion Model (Cond)**: A matching diffusion model implementation to NCS, fine-tuned on benchmark trajectories to address autonomous motion planning problems (Nichol and Dhariwal, 2021).
2. **Motion Planning Diffusion (MPD)**: The previous state-of-the-art for single-robot motion planning (Carvalho et al., 2023), this approach is extended to handle multi-agent settings for comparative analysis.
3. **Multi-robot Motion Planning Diffusion (MMD)**: A recently proposed method that integrates diffusion models with classical search techniques, generating collision-constrained multi-agent path planning solutions (Shaoul et al., 2024).



(*a*) Basic Maps. (*b*) Dense Maps.

Figure 5: Evaluation on random maps with three different numbers of robots. Gray bars represents the failure rate, and values on top of the bars indicate average path length per robot.

Symbolic test. We model two types of collision constraint tests. The first, ϕ_1 corresponds to collisions between agents, whereas ϕ_2 captures collisions between agents and obstacles in the map. We can express the collision-avoidance constraints as follows. First, for inter-agent separation, we require that for every pair of distinct agents i and i' and at every time step j, their positions are separated by at least a minimum distance $d_{\min}$ (which is defined as the sum of their radii):

$$\phi_1(\boldsymbol{x}) \stackrel{\text{def}}{=} \quad \forall i, i' \ (i \neq i'), \ \forall j \quad \|\boldsymbol{p}_i^j - \boldsymbol{p}_{i'}^j\|_2 \geq d_{\min}.$$

Second, to ensure agents do not collide with obstacles, we require that for each agent i at each time step j and for every obstacle k with radius r_k, the agent's position is at least $\boldsymbol{r}_k$ away from the obstacle's center $\boldsymbol{o}_k$:

$$\phi_2(\boldsymbol{x}) \stackrel{\text{def}}{=} \quad \forall i, \ \forall j, \ \forall k \quad \|\boldsymbol{p}_i^j - \boldsymbol{o}_k\|_2 \geq \boldsymbol{r}_k.$$

Here, i and i' index the agents, j denotes the time steps, $\boldsymbol{p}_i^j$ is the position of agent i at time j, and $\boldsymbol{o}_k$ is the position of obstacle k. As these constraints can be expressed in closed-form, we model $\Delta\phi$ directly from this, employing the augmented Lagrangian method to solve this projection (due to the highly non-convex nature of these constraints).

E.3. Morphometric Property Specification (Data Scarcity and Domain Generalization)

As mentioned in the main text, our dataset is generated by subsampling a single 3,000×3,000 pixel image into patches of size 64×64. We then upscale these patches to 256×256 images to increase the resolution for our generation task. The data is obtained from Chun et al. (2020), and we reiterate it contains only a small range of porosity values fall within the desired porosity ranges (e.g., at the porosity level P(%) 50, only 7% of the training data falls within a generous five percent error margin to either side), contributing to the challenging nature of this setting.

In the analysis of both natural and synthetic materials, heuristic metrics are commonly used to quantify microstructure features such as crystal shapes and void distributions. These measures provide qualitative insights into the fidelity of the synthetic samples relative to the training data. Here, we present the distributions of three microstructure descriptors, following the approach of Chun et al.

The results demonstrate that the explicit constraint enforcement in NSD yields microstructures that more faithfully replicate the ground truth. In contrast, the conditional model tends to produce certain features at frequencies that do not align with the training distribution. By integrating porosity and related constraints during the sampling process, NSD is able to generate a set of microstructures that is both more representative and accurate.

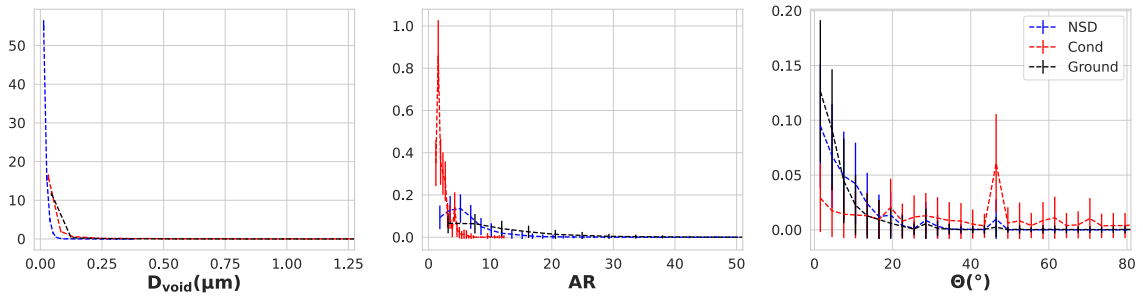


Figure 6: Morphometric parameter distributions comparing ground truth microstructures with those generated by the ***NSD*** and *Cond* models, evaluated using heuristic analysis.

Additional benchmarks. To supplement our evaluation, we compare to several domain specific approaches:

1. **Conditional Diffusion Model (Cond)**: A conditional diffusion model implementation modeled from Chun et al. (2020).
2. **Post-Processing (Post$^+$)**: A matching implementation to our diffusion model for NSD, with the projection steps omitted from the sampling process, except after the final step.
3. **Conditional + Post-Processing (Cond$^+$)**: The Cond model, but with the addition of a post-processing projection after the final step.

Symbolic test. We define a test function ϕ that measures the porosity of an image:

$$\phi(\boldsymbol{x}) \stackrel{\text{def}}{=} \Big(\sum_{i=1}^{n} \sum_{j=1}^{m} \big(\boldsymbol{x}^{i,j} < 0\big) \Big) = K,$$

where $\boldsymbol{x}^{i,j} \in [-1, 1]$ is the pixel value at row i and column j. Our desired constraint is that the porosity of the generated image $\boldsymbol{x}$ must equal a target value K. In our framework, this condition is used as a test that triggers the projection: if $\phi(\boldsymbol{x}) \neq 1$, a projection operator is applied to minimally adjust $\boldsymbol{x}$ so that the constraint is satisfied. This can be constructed using a top-k algorithm to return,

$$\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}^{i,j}} \sum_{i,j} \|\boldsymbol{y}^{i,j} - \boldsymbol{x}^{\boldsymbol{i},\boldsymbol{j}}\|$$

$$\text{s.t.} \quad \boldsymbol{y}^{i,j} \in [-1, 1], \quad \sum_{i=1}^{n} \sum_{j=1}^{m} \Big(\boldsymbol{y}^{\boldsymbol{i},\boldsymbol{j}} < 0\Big) = K$$

where K is the number of pixels that should be "porous". Because this program is convex, it serves as a certificate that the generated images comply with the prescribed porosity constraint.

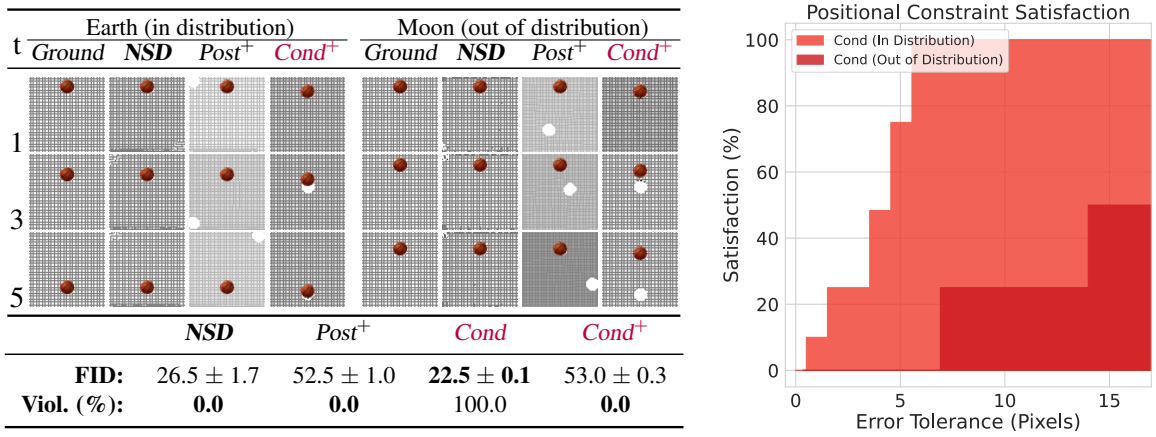


Figure 7: Physics-informed motion experimental results.

E.4. Physics-informed Motion (Data Scarcity and Domain Generalization)

For this setting, we generate a series of video frames depicting an object accelerating due to gravity. Here, the object's motion is governed by a system of ordinary differential equations (Eq. (9)), which our method directly integrates into the constraint set (ϕ). In addition to the complexity of the constraints, two key challenges are posed: **(1) Data Scarcity:** Our training data is based solely on Earth's gravity, yet our model is tested on gravitational forces from the Moon and other planets, where *there are no feasible training samples provided*, and, consequentially, **(2) Out-of-Distribution Constraints:** the imposed constraints are not represented in the training.

Figure 7 (left) highlights the results of our experiments; standard conditional diffusion models often produce objects that are misplaced within the frame, as evidenced by white object outlines in the generated samples and the reported constraint violations on the right side of the figure. Post-processing approaches correct positioning at the cost of significant image degradation. In contrast, our method **guarantees** *satisfaction of physical constraints while maintaining high visual fidelity*, producing samples that fully satisfy positional constraints. These results demonstrate that our approach generalizes to out-of-distribution physical conditions while ensuring strict compliance with governing physical laws.

For training, we begin by generating a dataset uniformly sampling various object starting points within the frame size [0,63]. For each data point, six frames are produced, depicting the objects movement as governed by the ODE in Equation (9). The velocity is initialized to $\mathbf{v}_0 = 0$. The diffusion models are trained on 1000 examples from this dataset, using a 90/10 training and testing split. The conditional model is implemented following Voleti et al. (2022), where we provide two frames illustrating the motion as conditioning. The model then infers future frames from these to produce the final videos.

Additional benchmarks. To supplement our evaluation, we compare to several domain specific approaches:

1. **Conditional Diffusion Model (Cond)**: A conditional diffusion model implementation as outlined by Voleti et al. (2022).

2. **Post-Processing (Post$^+$)**: A matching implementation to our diffusion model for NSD, with the projection steps omitted from the sampling process, except after the final step.
3. **Conditional + Post-Processing (Cond$^+$)**: The Cond model, but with the addition of a post-processing projection after the final step.

Symbolic test. We define a test function ϕ that checks whether the object's position in a frame meets the prescribed positional constraints given by Equations:

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \left(\mathbf{v}_t + \left(0.5 \times \frac{\partial \boldsymbol{v}_t}{\partial t}\right)\right) \tag{9a}$$

$$\mathbf{v}_{t+1} = \frac{\partial \boldsymbol{p}_t}{\partial t} + \frac{\partial \boldsymbol{v}_t}{\partial t}, \tag{9b}$$

Hence, we define our test function:

$$\phi(\boldsymbol{x}) \stackrel{\text{def}}{=} \big(\text{object position in } \boldsymbol{x} \text{ equals } \mathbf{p}_t\big)$$

In other words, the generated frame $\boldsymbol{x}$ is considered feasible if the object's position exactly matches the target position $\mathbf{p}_t$ computed by the dynamics model.

When $\phi(\boldsymbol{x}) \neq 1$, a projection operator is triggered to enforce the positional constraint. This projection proceeds in two steps. First, the object's current location is detected and its pixels are set to the maximum intensity (i.e., white), effectively removing the object from its original position while storing the indices of the object's structure. Second, the object is repositioned by mapping its stored pixel indices onto the center point corresponding to $\mathbf{p}_t$. If the frame already satisfies the positional constraint (i.e., $\phi(\boldsymbol{x}) = 1$), the projection leaves the image unchanged. Since this projection process is well-defined and convex, it provides a certificate that the generated frames comply with the prescribed positional constraints.

E.5. Safe Text Generation (Safety)

As language models become widely adopted for commercial and scientific applications, it is necessary that generated text adheres to safety constraints, preventing the production of harmful or toxic content. Hence, it is essential to provide **(1) Safety-Critical Outputs:** the adoption of constraint-aware methods is essential for these applications, especially considering recent examples of *toxic outputs encouraging self-harm or providing information which could be used to harm others* (Perez et al., 2022).

Table 3 highlights the results of our experiments, which evaluate language models on toxicity mitigation using prompts from the RealToxicityPrompts dataset (Gehman et al., 2020). Our method significantly improves control over generated content by enforcing strict toxicity constraints during inference. Compared to baseline models, such as GPT-2 and Llama 3.2, which exhibit high violation rates, *NSD achieves perfect constraint satisfaction across all toxicity thresholds*. Furthermore, our approach scales effectively, ensuring robust toxicity mitigation even at increasingly strict thresholds (denoted as τ).

Among the baselines, GPT-2 (124M) and LLaMA (1B) achieve the lowest perplexity scores; however, they frequently generate toxic content, leading to high violation rates. While GPT-2 + PPLM (345M) Dathathri et al. (2019) reduces violations across all toxicity thresholds, it fails to consistently prevent toxic generations and suffers from increased perplexity. MDLM (110M) exhibits higher perplexity than GPT-2, with a median PPL of 39.8, and although it moderately reduces toxicity violations compared to GPT-2, the rates remain significant. In contrast, NSD achieves *perfect constraint satisfaction* across all toxicity thresholds while maintaining sentence fluency.

	Model	Size	PPL		Viol (%)		
			Mean	Median	$\tau = 0.25$	$\tau = 0.50$	$\tau = 0.75$
Sentence Toxicity	GPT2	124M	19.1	17.6	36.3	23.8	16.2
	GPT2$_{PPLM}$	345M	47.6	37.2	15.2	8.1	4.3
	GPT2$_{\text{FUDGE}_{\lambda=2}}$	124M	26.46	18.79	31.5	19.7	12.7
	GPT2$_{\text{FUDGE}_{\lambda=9}}$	124M	81.84	19.22	30.6	19.6	11.7
	Llama 3.2	1B	**15.7**	**14.6**	34.9	27.8	23.1
	MDLM	110M	46.7	39.8	32.1	23.2	17.2
	NSD$_{\tau=0.25}$ (Ours)	110M	61.6	45.4	**0.0**	**0.0**	**0.0**
	NSD$_{\tau=0.50}$ (Ours)	110M	59.4	44.2	–	**0.0**	**0.0**
	NSD$_{\tau=0.75}$ (Ours)	110M	54.9	43.2	–	–	**0.0**

Table 3: Results for safe text generation at various toxicity levels τ.

Additional benchmarks. To supplement our evaluation, we compare to several domain specific approaches:

1. **GPT2**: Our model uses a GPT2 tokenizer and is roughly the same size as GPT2, so we add this as a point of comparison.
2. **Llama 3.2**: For comparison to state-of-the-art autoregressive models, we employ Llama 3.2, noting that this model is an order of magnitude larger than our diffusion model.
3. **Plug and Play Language Model (GPT2$_{PPLM}$)**: We utilize gradient-based guidance as proposed by Dathathri et al. (2019) to condition autoregressive generation against producing toxic outputs.
4. **Masked Diffusion Model (MDLM)**: A masked discrete language diffusion model implementation from Schiff et al. (2024).

Symbolic test. As ϕ cannot be explicitly modeled for general text toxicity quantification, we train a surrogate model to provide a differentiable scoring metric $\Delta\phi$. Hence, the constraint is assessed with respect to this learned metric, such that: $\Delta\phi(\boldsymbol{x}^\star) \leq \tau$, where τ is a tunable threshold that controls the degree of toxicity that's permissible (lower values resulting in less toxic output sequences). As the surrogate model for toxicity task, we use a GPT-Neo (1.3B) model, adapted for binary classification. We finetune this model on the Jigsaw toxicity dataset which includes multiple toxicity-related labels such as toxic, severe toxic, insult, etc. We consolidates these columns into a single binary target (toxic vs. non-toxic).

Appendix F. Missing Proofs

Proof (Theorem 1) By optimization theory of convergence in a convex setting, provided an arbitrary number of update steps t, $\boldsymbol{x}_t$ will reach the global minimum. Hence, this justifies the existence of $\bar{t}$ as at some iteration as $T \to \infty$,

$$\|\mathcal{U}_\theta(\boldsymbol{x}_t) - \Phi\|_2 \leq \|\rho - \Phi\|_2$$

which will hold for every iteration thereafter.

■

Proof (Theorem 2) For any update taken after convergence, consider a gradient update without the stochastic noise. There are two cases:

Case 1: Suppose that $\mathcal{U}_\theta(\boldsymbol{x}_t)$ is closer to the optimum than ρ. By the definition of ρ, this implies that $\boldsymbol{x}_t$ is infeasible. Moreover, a gradient step taken from an infeasible point will yield an update that is closer to the optimum than any point achievable from the feasible set. Hence, we obtain:

$$\mathit{Error}\Big(\mathcal{U}_\theta(\boldsymbol{x}_t)\Big) > \mathit{Error}\Big(\mathcal{P}_{\mathbf{C}}\big(\mathcal{U}_\theta(\boldsymbol{x}_t)\big)\Big). \tag{10}$$

Case 2: Suppose instead that $\mathcal{U}_\theta(\boldsymbol{x}_t)$ is equally close to the optimum as ρ. In this situation, either (1) $\boldsymbol{x}_t$ is already the closest feasible point to the optimum (i.e., $\boldsymbol{x}_t = \mathcal{P}_{\mathbf{C}}(\boldsymbol{x}_t)$), so that the error terms are equal, or (2) $\boldsymbol{x}_t$ is infeasible. In the latter case, the gradient step from $\boldsymbol{x}_t$ is equivalent in magnitude to that from the nearest feasible point, but, by convexity, the triangle inequality ensures that the error from starting at an infeasible point exceeds that from starting at the feasible projection. Thus, Equation (10) holds in all cases. Finally, when the stochastic noise (sampled from a zero-mean Gaussian) is incorporated, taking the expectation over the update yields:

$$\mathbb{E}\Big[\mathit{Error}\Big(\mathcal{U}_\theta(\boldsymbol{x}_t)\Big)\Big] \geq \mathbb{E}\Big[\mathit{Error}\Big(\mathcal{P}_{\mathbf{C}}\big(\mathcal{U}_\theta(\boldsymbol{x}_t)\big)\Big)\Big].$$

■