Context-Guided Diffusion for Out-of-Distribution Molecular and Protein Design

Leo Klarner [1] **Tim G. J. Rudner** [2] **Garrett M. Morris** [1] **Charlotte M. Deane** [1] **Yee Whye Teh** [1]

Abstract

Generative models have the potential to accelerate key steps in the discovery of novel molecular therapeutics and materials. Diffusion models have recently emerged as a powerful approach, excelling at unconditional sample generation and, with data-driven guidance, conditional generation within their training domain. Reliably sampling from high-value regions beyond the training data, however, remains an open challenge—with current methods predominantly focusing on modifying the diffusion process itself. In this paper, we develop context-guided diffusion (CGD), a simple plug-and-play method that leverages unlabeled data and smoothness constraints to improve the out-of-distribution generalization of guided diffusion models. We demonstrate that this approach leads to substantial performance gains across various settings, including continuous, discrete, and graph-structured diffusion processes with applications across drug discovery, materials science, and protein design.

1. Introduction

The central goal of molecular discovery is to identify novel compounds with desirable functional properties. The immense size of the underlying search spaces—up to 10^{60} for drug-like small molecules (Bohacek et al., 1996) and 20^N for N-length protein sequences (Maynard Smith, 1970)—renders this a challenging combinatorial optimization problem. Considering the substantial cost of synthesizing and validating candidate compounds experimentally, the question of how to efficiently navigate these search spaces to locate high-value subsets lies at the core of modern molecular design (Gómez-Bombarelli et al., 2018; Sanchez-Lengeling & Aspuru-Guzik, 2018; Bilodeau et al., 2022).

[1]University of Oxford, UK [2]New York University, New York, USA. Correspondence to: Leo Klarner <leo.klarner@stats.ox.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

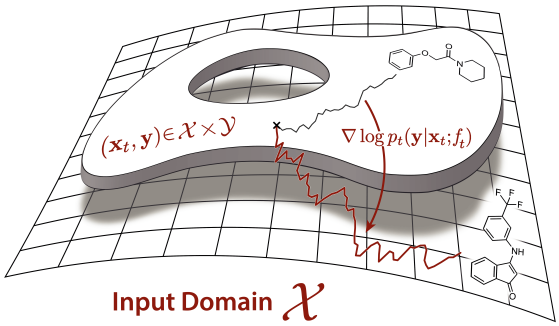


Figure 1: Guidance models that generalize poorly under distribution shifts can be a major performance bottleneck for property-guided diffusion models. We introduce a guidance model regularizer that improves generalization under distribution shifts and enables *context-guided diffusion* (CGD). We show that CGD leads to conditional sampling processes that consistently generate novel, high-value molecules (**red**).

Deep generative models have the potential to accelerate this process by capturing and abstracting key structural properties of their input domain (Sanchez-Lengeling & Aspuru-Guzik, 2018; Bilodeau et al., 2022). In recent years, denoising diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020; Karras et al., 2022) have emerged as the method of choice, displaying impressive performance in conditional and unconditional image generation tasks (Dhariwal & Nichol, 2021; Saharia et al., 2022; Rombach et al., 2022; Zhang et al., 2023), as well as applications across chemistry (Jing et al., 2022; Corso et al., 2023) and biology (Watson et al., 2023; Abramson et al., 2024).

While diffusion models excel at capturing complex, multimodal densities (Kadkhodaie et al., 2024), accurately modeling the training distribution is only the first step. To fully leverage diffusion models for applied molecular optimization, we require carefully designed guidance functions that steer the sample generation process toward compounds with desirable properties (Weiss et al., 2023; Gruver et al., 2023; Lee et al., 2023). However, when labeled data for training these guidance functions is scarce and only available for a biased and unrepresentative subset of the input domain, as is often the case in practice, overconfident guidance signals risk steering the generative process toward false-positive regions of chemical or protein sequence space (Figure 1).

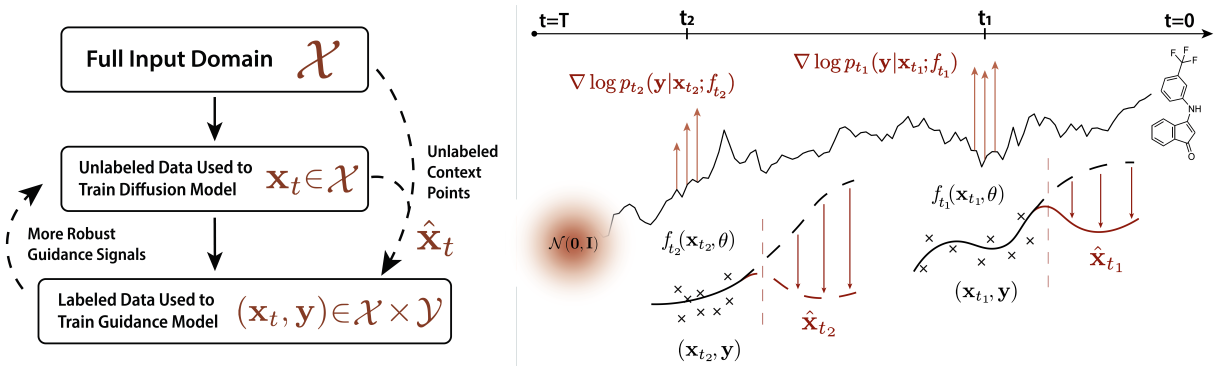


Figure 2: *Context-guided diffusion* leverages unlabeled context data to combine signals from labeled training data with structural information of the broader input domain **(left)**. Specifically, we construct a data- and noise scale-dependent guidance model regularizer that encourages smooth gradients, mean reversion, and high predictive uncertainty on out-of-distribution (OOD) inputs, allowing the conditional denoising process under a context-guided diffusion model to focus on promising near-OOD subsets of chemical and protein sequence space **(right)**.

In this paper, we propose *context-guided diffusion* (CGD)—a simple plug-and-play method that leverages unlabeled data and smoothness constraints to improve the out-of-distribution generalization of property-guided diffusion models. Specifically, we construct a guidance model regularizer that leverages unlabeled context data to train guidance models that both (i) fit the training data well and (ii) exhibit high uncertainty and smooth gradients in out-of-distribution (OOD) regions of the input domain (Figure 2). We show that using the resulting context-aware guidance model to perform context-guided diffusion leads to conditional sampling processes that consistently generate novel, near-OOD molecules with desirable properties.

CGD is straightforward to implement, requires no changes to the architecture of the guidance model, and adds no computational overhead at sampling time. Crucially, it is agnostic to the underlying generative process and thus compatible with latent (Rombach et al., 2022; Xu et al., 2023), equivariant (Hoogeboom et al., 2022), Riemannian (De Bortoli et al., 2022; Huang et al., 2022), and constrained (Lou & Ermon, 2023; Fishman et al., 2023a;b) diffusion models.

We demonstrate the versatility of our approach by applying it to the design of small molecules with graph-structured diffusion processes (Lee et al., 2023), the generation of novel materials with equivariant diffusion models (Weiss et al., 2023), and the optimization of discrete protein sequences with categorical diffusion (Gruver et al., 2023). We find that CGD consistently outperforms standard guidance models, as well as more sophisticated pre-training and domain adaptation techniques, and that it enables more reliable generation of novel samples from high-value subsets of chemical and protein sequence space.

The code for our experiments can be accessed at:
`https://github.com/leojklarner/context-guided-diffusion`

2. Guided Diffusion Models

We start by providing a brief overview of standard unconditional diffusion models, following Song et al. (2020), and then discuss conditional sampling schemes, focusing on classifier guidance as introduced in Sohl-Dickstein et al. (2015), Song et al. (2020) and Dhariwal & Nichol (2021).

Unconditional Diffusion Models. Denoising diffusion models progressively add noise to a data distribution p_0 until it approaches a tractable reference distribution p_T. This forward noising process is governed by a stochastic differential equation (SDE) (Song et al., 2020), defined as

$$\mathrm{d}\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, t) + g(t)\mathrm{d}\mathbf{B}_t, \tag{1}$$

where $\mathbf{X}_0 \sim p_0$ is sampled from the data distribution and $\mathbf{B}_t$ represents the d-dimensional Brownian motion. A common choice is the VP-SDE (Sohl-Dickstein et al., 2015; Ho et al., 2020), with drift coefficient $\mathbf{f}(\mathbf{X}_t, t) = -\frac{1}{2}\beta_t \mathbf{X}_t$ and diffusion coefficient $g(t) = \sqrt{\beta_t}$, where the noise scale $\beta_t > 0$ determines the amount of corruption. Over time steps $t \in [0, T]$, this SDE gradually noises $\mathbf{X}_t \sim p_t$ until it converges to the reference distribution $p_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

To generate new samples from p_0, diffusion models leverage the fact that the forward noising process defined by Equation (1) admits a time-reversal (Anderson, 1982; Haussmann & Pardoux, 1986). For the VP-SDE, this process is given by

$$\mathrm{d}\mathbf{X}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{X}_t + \nabla \log p_t\left(\mathbf{X}_t\right) \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t, \tag{2}$$

which flows backward in time. $\mathbf{X}_T \sim p_T$ is sampled from the reference distribution and incrementally denoised until it reflects p_0. We have used the shorthand $\nabla \log p_t\left(\mathbf{X}_t\right)$ to denote $\nabla_{\mathbf{x}} \log p_t\left(\mathbf{x}\right)|_{\mathbf{x}=\mathbf{X}_t}$ and will continue using this shorthand in the remainder of the text.

The gradient of the log-density $\nabla \log p_t(\mathbf{x}_t)$ in Equation (2) is estimated by a time-dependent score network $s_\psi(\mathbf{x}_t, t)$ using techniques from score matching (Hyvärinen & Dayan, 2005). Once trained, s_ψ can be used to generate samples from the data distribution p_0 by simulating the reverse-time SDE with a range of numerical solvers (Song et al., 2020).

Guided Diffusion Models. In many applications, we may wish to condition the generative process of diffusion models to produce samples with specific properties $\mathbf{y}$, such as images of a certain class or molecules with a desirable pharmacological effect. That is, we want to modify the denoising process in Equation (2) to incorporate the conditioning information $\mathbf{y}$, and sample from the conditional distribution $p_0(\mathbf{x}_0 \,|\, \mathbf{y})$ rather than the unconditional data distribution $p_0(\mathbf{x}_0)$. The corresponding reverse-time SDE is given by

$$\mathrm{d}\mathbf{X}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{X}_t + \nabla \log p_t\left(\mathbf{X}_t \mid \mathbf{y}\right) \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t,$$

differing from Equation (2) only in the need for a conditional score function $\nabla \log p_t(\mathbf{X}_t \mid \mathbf{y})$, which can be expressed as

$$\begin{aligned} \nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}) &= \nabla \log \frac{p_t(\mathbf{y} \mid \mathbf{X}_t) p_t(\mathbf{X}_t)}{Z} \\ &= \nabla \log p_t(\mathbf{y} \mid \mathbf{X}_t) + \nabla \log p_t(\mathbf{X}_t), \end{aligned} \tag{3}$$

since Z is a normalization constant independent of $\mathbf{X}_t$.

To approximate $\nabla \log p_t(\mathbf{y} \mid \mathbf{X}_t)$, a time-dependent discriminative *guidance model* $f_t(\mathbf{x}_t; \theta)$ is trained on a labeled dataset $\mathcal{D} = \cup_{t=1}^{T} \{(\mathbf{x}_t^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$ with inputs corrupted by the same forward noising process as the diffusion model. Once trained, the guidance model can be incorporated into a *guidance function* $p(\mathbf{y} \mid \mathbf{X}_t; f(\cdot;\theta))$ that is used to compute the *conditional scores* $\nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}; f(\cdot;\theta))$, which in turn steer the reverse-time SDE toward samples with a high predicted likelihood of observed label $\mathbf{y}$ via

$$\mathrm{d}\mathbf{X}_t = -\beta_t \left\{ \frac{1}{2}\mathbf{X}_t + \nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}; f_t(\cdot;\theta)) \right\} \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t.$$

Alternative approaches such as classifier-free guidance (Ho & Salimans, 2022) estimate the conditional scores by passing the conditioning information to the score network

$$\nabla \log p_t(\mathbf{X}_t \mid \mathbf{y}) \approx s_\phi(\mathbf{x}_t, \mathbf{z}, t),$$

either in the form of class labels $\mathbf{z} = \mathbf{y}$ (Ho & Salimans, 2022) or pre-trained embeddings $\mathbf{z} \sim p(\mathbf{z}|\mathbf{y})$ (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022). However, these methods' inability to provide an explicit *guidance function gradient estimate* $\nabla \log p_t(\mathbf{y} \mid \mathbf{X}_t)$ limits their applicability to regression-based optimization problems frequently encountered in molecular and protein design (Lee et al., 2023; Weiss et al., 2023; Gruver et al., 2023), where the objective is to maximize or minimize $\mathbf{y}$ without prior knowledge of its optimal value.

3. Context-Guided Diffusion Models

In this section, we present *context-guided diffusion* (CGD)—a method that produces more robust conditional gradient estimates that consistently generate novel samples with improved properties. We first introduce the prevailing approach to training guidance models and highlight its limitations. We then present a domain-informed regularization term designed to bias guidance model training toward functions that both (i) fit the training data well and (ii) revert to domain-appropriate behavior in out-of-distribution settings. Finally, we describe how to incorporate context-aware guidance models into guided diffusion to generate out-of-distribution samples with desirable properties.

3.1. Standard Guidance Model Training

We consider a standard supervised learning setting with diffusion timesteps $t = 0, ..., T$ and N i.i.d. data realizations $\mathcal{D}_t = \{(\mathbf{x}_t^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$ of the noised inputs $\mathbf{x}_t \in \mathcal{X}$ and targets $\mathbf{y} \in \mathcal{Y}$. Furthermore, we define the guidance model $f_t(\cdot\,;\theta) \doteq \theta_L h_t(\cdot\,;\theta_h)$ with $\theta \doteq \{\theta_h, \theta_L\}$ as the composition of an embedding model $h_t(\cdot\,;\theta_h) : \mathcal{X} \to \mathbb{R}^d$ and a linear output head $\theta_L : \mathbb{R}^d \to \mathcal{Y}$.

Conventional supervised learning techniques for training guidance models define the diffusion timestep-dependent negative log-likelihood objective

$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T}) \doteq -\sum_{n=1}^{N} \log p_\mathcal{T}(\mathbf{y}^{(n)} \mid \mathbf{x}_\mathcal{T}^{(n)}; f_\mathcal{T}(\cdot;\theta))$$

where $\mathcal{T} \sim p_\mathcal{T}(t)$ is a randomly sampled timestep. The guidance model parameters θ are optimized by minimizing the L_2-regularized objective function

$$\bar{\mathcal{L}}(\theta, \mathcal{D}_\mathcal{T}) \doteq \mathbb{E}_{p_\mathcal{T}}\left[\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T})\right] + \frac{1}{2\lambda}||\theta||_2^2, \tag{4}$$

where $\lambda \in \mathbb{R}_+$ is chosen as a hyperparameter.

Unfortunately, this approach is only able to operate over subsets of the input domain $\mathcal{X}$ for which labeled data is available. When labels are scarce, as is the case in many practical settings, models trained with standard supervised learning objectives are prone to generating incorrect and overconfident predictions when exposed to distribution shifts (Ovadia et al., 2019; Liu et al., 2020; Van Amersfoort et al., 2020; Koh et al., 2021; Band et al., 2021; Rudner et al., 2021; Tran et al., 2022; Rudner et al., 2022; Klarner et al., 2023). This issue is exacerbated by the fact that guidance models are trained on noised data points: Since neural networks are both highly sensitive to noise perturbations (Szegedy et al., 2013; Goodfellow et al., 2014; Ho & Salimans, 2022) and able to overfit to fully corrupted inputs (Zhang et al., 2021), the errors of systematically overconfident gradient signals may accumulate over hundreds of denoising steps.

Algorithm 1 One guidance model training iteration with the context-guided regularization scheme from Equation (9).

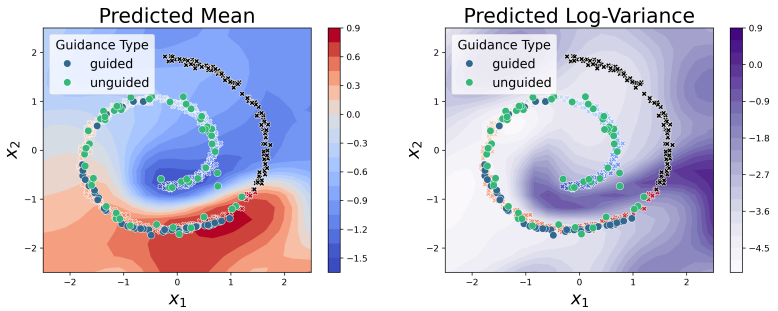
Require:
Data and context batch: $(\mathbf{x}_0, \mathbf{y}), \hat{\mathbf{x}}_0$
Diffusion time steps: $t, t' \sim \mathcal{U}(0, T)$
Guidance model: $f_t(\cdot; \theta)$
Regularizer: $D_{\mathcal{M}}(f_t(\cdot;\theta), Q_t(\cdot))^2$
Hyperparameters: $\sigma_t, \tau_t, \lambda$

$\mathbf{x}_t \leftarrow \text{NoisingProcess}(\mathbf{x}_0, t)$
$\bar{\mathbf{y}} \leftarrow f_t(\mathbf{x}_t; \theta)$
$\bar{\mathcal{L}} \leftarrow -\log p_t(\mathbf{y} \mid \mathbf{x}_t; \bar{\mathbf{y}}) + \frac{1}{2\lambda}||\theta||_2^2$

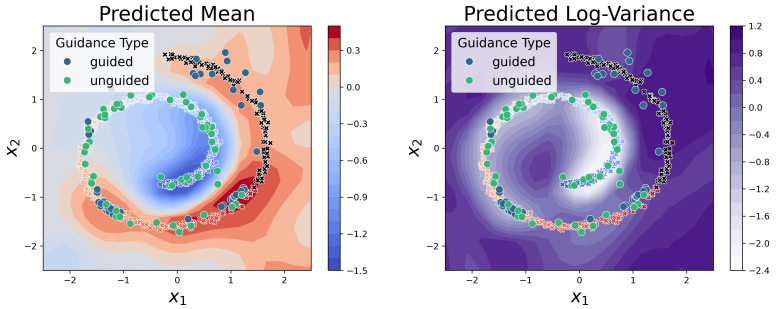
$\hat{\mathbf{x}}_{t'} \leftarrow \text{NoisingProcess}(\hat{\mathbf{x}}_0, t')$
$\hat{\mathbf{y}} \leftarrow f_{t'}(\hat{\mathbf{x}}_{t'}; \theta)$
$R \leftarrow \sum_{j=1,2} D_{\mathcal{M}}(\hat{\mathbf{y}}^j, Q^j_{t'}(\hat{\mathbf{x}}_{t'}))^2$

$\theta \leftarrow \text{Update}(\theta, \bar{\mathcal{L}} + R)$

(a) An algorithmic overview of the proposed regularization scheme.



(b) Predictions and samples from an L_2-regularized guidance model.



(c) Predictions and samples from a context-aware guidance model.

Figure 3: Panel (a) presents an algorithmic perspective on how the proposed regularization scheme fits into the training loop of a standard guidance model. Panels (b) and (c) illustrate how the regularization scheme affects the functions a guidance model learns. Two regression models with different regularizers are trained on a low-label subset of an illustrative dataset (color-coded crosses). However, without additional information about the input domain, the predictions of the L_2-regularized model (b) exhibit poor generalization and miscalibrated uncertainty estimates when evaluated in new, unseen regions. In contrast, the context-aware model (c) is able to leverage unlabeled data sampled uniformly from $[-2.5, 2.5]^2 \subset \mathbb{R}^2$ to generate more accurate and better-calibrated predictions. The corresponding context-guided denoising process generates samples that recover a held-out, high-label test set (black crosses). Full experimental details are presented in Appendix B.1.

3.2. Context-Aware Guidance Models

To improve the performance and reliability of guidance models in this setting, we propose a family of diffusion timestep-dependent regularizers designed to favor parameters θ that encode desirable model behaviors on out-of-distribution data. While ideal model behaviors under distribution shifts are often application-dependent, we design a general-purpose regularizer—applicable across domains—that encourages uncertain predictions in areas of the input domain with little or no signal from the labeled data. This regularizer maximizes sensitivity at the expense of specificity and enables the conditional denoising process to focus on regions in data space that are near the training data and have the highest likelihood of containing molecules with improved properties.

To encode this behavior, we construct a data- and timestep-dependent distribution $Q_t(\cdot)$ that is explicitly designed to exhibit high predictive uncertainty on a problem-specific, out-of-distribution context set $\hat{\mathbf{x}}_{\mathcal{C}} \doteq \{\mathbf{x}_{\mathcal{C}}^{(i)}\}_{i=1}^{N_{\mathcal{C}}}$.

More specifically, we consider regularizers of the form

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) \doteq \mathbb{E}_{p_{\hat{\mathbf{X}}_t}}[D_{\mathcal{M}}(f_t(\hat{\mathbf{x}}_t;\theta), Q_t(\hat{\mathbf{x}}_t))^2], \quad (5)$$

where $p_{\hat{\mathbf{X}}_t}$ is a uniform distribution over noised context batches $\hat{\mathbf{x}}_t$ of size M (with $M \ll N_{\mathcal{C}}$).

$$D_{\mathcal{M}}(f_t(\hat{\mathbf{x}}_t;\theta), Q_t(\hat{\mathbf{x}}_t))^2 \quad (6)$$
$$\doteq (f_t(\hat{\mathbf{x}}_t;\theta) - m_t(\hat{\mathbf{x}}_t))^\top K_t(\hat{\mathbf{x}}_t)^{-1}(f_t(\hat{\mathbf{x}}_t;\theta) - m_t(\hat{\mathbf{x}}_t))$$

is the squared Mahalanobis distance between noised model predictions $f_t(\hat{\mathbf{x}}_t;\theta)$ and a data- and timestep-dependent distribution $Q_t(\hat{\mathbf{x}}_t)$ with mean $m_t(\hat{\mathbf{x}}_t)$ and covariance $K_t(\hat{\mathbf{x}}_t)$. Here, $m_t(\hat{\mathbf{x}}_t)$ specifies the desired out-of-distribution behavior, and $K_t(\hat{\mathbf{x}}_t) \in \mathbb{R}^{M \times M}$ allows us to encode additional information about the structure of the input domain. Specifically, $K_t(\hat{\mathbf{x}}_t) \neq \mathbf{I}$ controls how strongly the guidance model predictions $f_t(\hat{\mathbf{x}}_t;\theta)$ can vary between similar context points, allowing us to enforce smoother gradient estimates and more stable guidance signals in regions of data space with limited label signal.

This covariance function can be defined in terms of any valid, domain-appropriate similarity metric. In the following, we present a general approach to constructing K_t from model embeddings. Specifically, we consider the neural network embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M \times d}$ of all molecules in the context batch $\hat{\mathbf{x}}_t$ and use their inner product to construct

$$K_t(\hat{\mathbf{x}}_t) \doteq \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}, \tag{7}$$

where σ_t is a covariance scale parameter and τ_t is a diagonal offset parameter. These hyperparameters allow us to calibrate the behavior of our regularizer, with τ_t determining how closely the predictions have to match the mean function $m_t(\hat{\mathbf{x}}_t)$ and σ_t determining the strength of the smoothness constraints placed on $f_t(\hat{\mathbf{x}}_t; \theta)$.

This approach is agnostic to how the embeddings $h_t(\hat{\mathbf{x}}_t, \phi)$ are derived, so long as they capture relevant structural properties of the input domain. We find that a relatively simple approach is already sufficient to produce sizeable performance gains in the applications we consider here. Specifically, we use a fixed set of randomly initialized parameters ϕ to obtain $h_t(\hat{\mathbf{x}}_t, \phi)$ from the model's embedding trunk. This is motivated by the observation that neural networks with suitable inductive biases, such as permutation- or roto-translational equivariance in Sections 5.1 and 5.2, can already produce robust and informative representations in a randomly initialized state (Kipf & Welling, 2016).

Having constructed the covariance matrix $K_t(\hat{\mathbf{x}}_t)$, we specify the mean function $m_t(\hat{\mathbf{x}}_t)$ as follows. Letting $f_t(\cdot; \theta)$ be a standard multi-output regression model consisting of a learned mean function $f_t^1(\hat{\mathbf{x}}_t; \theta)$ and, optionally, a learned variance function $f_t^2(\hat{\mathbf{x}}_t; \theta)$, we wish for the mean predictions to revert to the mean of the training labels and for the variance to be high when the model is evaluated on points that are meaningfully distinct from the training data. To encode this desired behavior, we define the regularizer

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) \doteq \mathbb{E}_{p_{\hat{\mathbf{X}}_t}} \left[\sum\nolimits_{j=1}^{2} D_{\mathcal{M}}(f_t^j(\hat{\mathbf{x}}_t; \theta), Q_t^j(\hat{\mathbf{x}}_t))^2 \right], \tag{8}$$

where $m_t^1(\hat{\mathbf{x}}_t) \doteq \frac{1}{N} \sum_{n=1}^{N} y^{(n)}$ for the training data labels $y^{(n)}$, $m_t^2(\hat{\mathbf{x}}_t) \doteq \boldsymbol{\sigma}_0^2$, for a target variance parameter $\boldsymbol{\sigma}_0^2$, and $p_{\hat{\mathbf{X}}_t}$ is defined as in Equation (5).

Adding this regularizer to the supervised learning loss from Equation (4), we arrive at the modified training objective

$$\mathcal{L}^*(\theta, \mathcal{D}_{\mathcal{T}}) = \mathbb{E}_{p_{\mathcal{T}}}[\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}, \mathcal{T}) + R(\theta, f_{\mathcal{T}}, \mathcal{T}, p_{\hat{\mathbf{X}}_{\mathcal{T}}})], \tag{9}$$

where both expectations (over $p_{\mathcal{T}}$ and $p_{\hat{\mathbf{X}}_t}$) can be estimated via simple Monte Carlo estimation.

Implementation. The proposed regularizer R does not require any changes to the guidance model itself and can be easily integrated into standard guidance model training pipelines: At each iteration, we sample a mini-batch $\hat{\mathbf{x}}_t$ of size M from the context set $\hat{\mathbf{x}}_{\mathcal{C}}$ and perturb it with the same noising process as the unconditional diffusion model in Equation (1). We then perform a gradient-free forward pass of the embedding model to compute $h_t(\hat{\mathbf{x}}_t, \phi)$ and construct the covariance matrix $K_t(\hat{\mathbf{x}}_t)$. Finally, we get the predictions of the guidance model $f_t(\hat{\mathbf{x}}_t; \theta)$ on the context batch to compute the full regularization term in Equation (8). An algorithmic view of this process is shown in Figure 3.

As the risk of miscalibrated gradient estimates is particularly high in the earlier stages of the sample generation process—where highly diffused input points are associated with the labels of the corresponding un-noised points—it is important to ensure that our regularizer is able to take the noise level β_t into account. Specifically, we increase τ_t with the same schedule as the noise levels β_t to penalize overconfident predictions as context points approach the reference distribution $p_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Similarly, as the distinction between in-distribution and out-of-distribution data points becomes less meaningful at larger noise levels, we decrease σ_t with an inverted schedule.

3.3. Generation via Context-Guided Diffusion

Once the context-aware guidance model $f_t(\cdot; \theta)$ has been trained, we can use it to steer the denoising process towards out-of-distribution samples that are most likely to exhibit desired properties. To do so, we use the context-aware guidance model to specify a context-aware guidance function $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot; \theta))$. This function defines a distribution—conditioned on both the training data and the context set—that reflects the probability of obtaining the desired properties given $\mathbf{x}_t$ and allows integrating signals from labeled data points along with structural knowledge about the broader input domain into sampling. Details about guidance function specification are provided in Appendix A, and an illustrative example that showcases the effect of the proposed regularizer on the guidance model (and the resulting guided diffusion process) is shown in Figure 3.

Finally, the context-aware guidance function can be seamlessly integrated into existing guided diffusion model frameworks by replacing the standard guidance function at sampling time. This modular, plug-and-play approach ensures that our method is independent of the specific diffusion model architecture and generative process and requires minimal implementation overhead. Importantly, like conventional guided diffusion, our method does not necessitate any (re-)training or fine-tuning of the diffusion model itself, making it a convenient out-of-the-box tool for guiding the generation process of a pre-trained model.

4. Related Work

Guided Diffusion for Molecular Design. Guiding the denoising process of diffusion models with gradients from a classifier or conditional score network is an effective approach that has been applied to a broad range of scientific problems, such as the generation of desirable protein folds (Watson et al., 2023; Ingraham et al., 2023) and molecular conformations (Hoogeboom et al., 2022; Peng et al., 2023; Guan et al., 2023). Similarly to our approach, recent work has explored property-guided diffusion models for molecular optimization with a particular focus on out-of-distribution sample generation. For instance, Lee et al. (2023) proposed a modification of the unconditional denoising process that, in combination with standard supervised guidance, generated more diverse small molecules with improved properties. Likewise, Weiss et al. (2023) used a guided diffusion model to optimize the electronic properties of polycyclic aromatic systems, demonstrating generalization to unseen heterocycles. Furthermore, Gruver et al. (2023) explored property-guided discrete diffusion for multi-objective protein sequence optimization, balancing naturalness with improved properties through a grid of sampling hyperparameters. However, all of these approaches primarily focus on modifying the unconditional sampling process itself and rely on standard supervised learning techniques for guidance model training. Our method is orthogonal to these contributions and aims to further improve the performance of property-guided diffusion models across application domains and generative processes.

Regularization with Unlabeled Data. Using unlabeled data to improve the performance of predictive models is a well-established paradigm that underlies many successful semi-supervised (Kingma et al., 2014; Laine & Aila, 2016; Berthelot et al., 2019) and self-supervised (Brown et al., 2020; Bommasani et al., 2021) methods. This includes a range of unsupervised domain adaptation and generalization techniques that use unlabeled data to mitigate the adverse effect of distribution shifts (Tzeng et al., 2015; Sun & Saenko, 2016; Ganin et al., 2016; Kang et al., 2019; Li et al., 2020). However, these methods predominantly focus on aligning model embeddings and are, unlike our approach, unable to directly influence a model's predictions. The method most closely related to our approach is the regularization scheme proposed in Rudner et al. (2023). However, in contrast to Rudner et al. (2023), who only consider supervised image classification tasks, our approach is specifically designed for the conditional generation of molecular structures. Key methodological developments in this regard include the introduction of a factorized regularization term that is suitable for use with regression models, as well as an explicit timestep-dependence that accounts for the noising process of the underlying diffusion model.

5. Empirical Evaluation

We compare our method to both standard guidance models and sophisticated pre-training and domain adaptation techniques across a range of experimental settings. Specifically, we demonstrate the versatility of context-guided diffusion by applying it to the design of small molecules with graph-structured diffusion processes (Section 5.1), to the generation of novel materials with equivariant diffusion models (Section 5.2), and to the optimization of discrete protein sequences with categorical diffusion models (Section 5.3). We observe that context-guided diffusion leads to improved conditional sampling processes that consistently and substantially outperform existing methods across diffusion model types and application domains.

5.1. Graph-Structured Diffusion for Small Molecules

To evaluate our method in the context of graph-structured diffusion processes for small molecule generation, we follow the framework of Lee et al. (2023) and consider a discrete diffusion process over the space of molecular graphs. Defining a graph $\mathbf{g}_t$ as a tuple of node feature and adjacency matrices $\mathbf{g}_t = (\mathbf{x}_t, \mathbf{a}_t)$, the forward and reverse processes are given by a joint system of SDEs (Jo et al., 2022). To encourage the exploration of novel chemical space, Lee et al. (2023) condition the denoising process on a hyperparameter $\gamma \in [0, 1)$ that shrinks the unconditional gradients

$$\begin{aligned}\nabla \log p_t(\mathbf{g}_t|\gamma) &= \nabla \log p_t(\mathbf{g}_t) + \nabla \log p_t(\gamma|\mathbf{g}_t) \\ &= (1 - \sqrt{\gamma})\, \nabla \log p_t(\mathbf{g}_t).\end{aligned} \tag{10}$$

In addition, the denoising process is conditioned on desirable molecular properties via a standard supervised guidance model as outlined in Sections 2 and 3.1. This model is trained to predict the labels $\mathbf{y} \in [0, 1]$, defined as a composite of the synthetic accessibility (SA; Ertl & Schuffenhauer, 2009), drug-likeness (QED; Bickerton et al., 2012), and QuickVina docking scores (Alhossary et al., 2015), described in further detail in Appendix B.2. In this setting, larger labels denote more desirable properties.

As the docking scores are protein pocket-dependent, we repeat all experiments across the five different targets chosen by Lee et al. (2023), namely PARP1, FA7, 5HT1B, BRAF and JAK2. We closely follow their experimental setup and train guidance models on a subset of 250 000 small molecules from the ZINC database (Irwin et al., 2012). Specifically, we pre-compute the labels $\mathbf{y}$ for every target and use them to split the data into a low-property training and a high-property validation set. This label split allows us to select regularization hyperparameters that maximize the ability of the guidance models to generalize well to novel, high-value regions of chemical space, serving as a proxy for the desired behavior at sampling time.

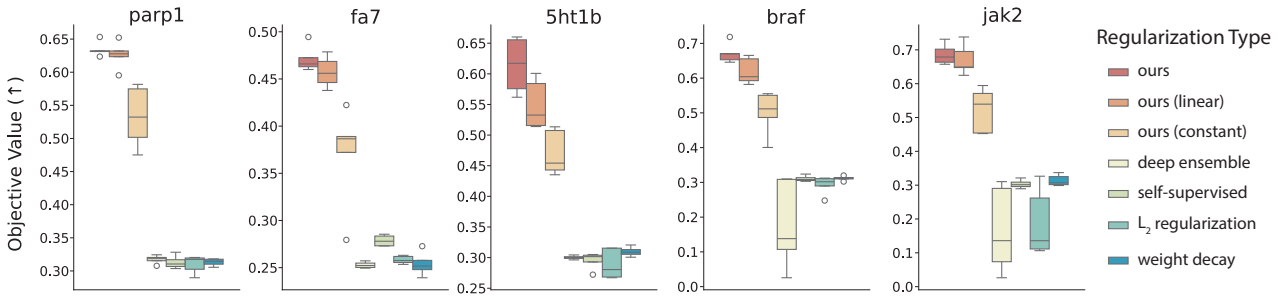


Figure 4: Comparison of the small molecules generated with different guided diffusion models across five distinct protein targets. Objective values (↑) are normalized with respect to the highest score of the held-out high-property validation set and averaged across five independent training and sampling runs with different random seeds.

To construct an appropriate context set for use with our regularizer, we sample an additional 500 000 unlabeled small molecules from the ZINC database. Following Section 3.2, we specify a predictive distribution $Q_t(\hat{\mathbf{x}}_t)$ that reverts to the training set mean and high uncertainty on out-of-distribution context points. We compare this approach to a range of alternative regularization schemes, including weight decay (used in Lee et al. (2023)), explicit L_2 regularization, and an ensemble of independently trained networks (Lakshminarayanan et al., 2017). To facilitate a fair comparison of these methods, we ensure that all training parameters, except regularization type and strength, are kept constant across experiments. After selecting the optimal hyperparameters using the high-property validation set, all models are independently retrained with five different random seeds. We refer to Appendix B.2 for full implementational details, hyperparameter ranges, additional results, and sanity checks.

We additionally evaluate the performance of other approaches that are able to make use of unlabeled data. Specifically, we pre-train a guidance model on all molecules from our context set with a self-supervised denoising objective (Zaidi et al., 2023) and then fine-tune it on the labeled data, using weight decay. Furthermore, we consider a range of unsupervised domain adaptation and generalization techniques, namely DeepCoral (Sun & Saenko, 2016), domain-adversarial neural networks (Ganin et al., 2016), and domain confusion (Tzeng et al., 2015), and present the results for these methods in Appendix B.2.

Adhering to the evaluation protocol of Lee et al. (2023), we use the retrained guidance models to generate 3000 molecules each and derive the corresponding labels with the provided docking and evaluation pipeline. As in Lee et al. (2023), we only retain compounds that pass synthetic accessibility and drug-likeness thresholds and display a maximum ECFP4-based Tanimoto similarity (Rogers & Hahn, 2010) of less than 0.4 to any molecule in the training set, before reporting the top 5% of the remaining docking scores.

To put these absolute values into perspective, we normalize them so that 1 corresponds to the highest docking score in the high-property validation set. The results are reported in Figure 4 and demonstrate that our method consistently improves the performance of property-guided diffusion models across different protein targets. The full distributions of all components of the objective function, the results of models trained with label noise, as well as versions normalized by heavy atom count, are presented in Appendix B.2 and display identical trends. While ensembling and pre-training techniques perform best on in-distribution tasks (see Appendix B.2), we find that they alone are not sufficient to overcome the limitations of standard supervised training/fine-tuning techniques. However, we note that context guidance is an orthogonal approach that is straightforward to use in combination with these methods.

To investigate how important the timestep-dependence of our method is to its empirical performance, we carry out a series of ablations in which the regularization hyperparameters σ_t and τ_t either (i) mimic the β_t schedule of the noising process, (ii) change linearly over time, or (iii) stay constant across all noise scales. We observe a strict performance drop across these settings, indicating that adapting the regularizer to the noising process is highly beneficial (Figure 4).

Furthermore, we investigate the impact that the size and composition of our context set have on the performance of our method by comparing models trained with the full context set to models trained with randomly sampled subsets of size 10% and 1%, respectively. As expected, we observe a strong decrease in performance, with the results of models trained with the smallest context set (1%) reverting to that of standard weight decay. Additionally, we select the 10% of compounds in the original context set that are either most or least similar to the labeled training data and find that using a much smaller set of more similar, near-OOD molecules is able to match the performance of the full context set. Full details and additional results are presented in Table 6.

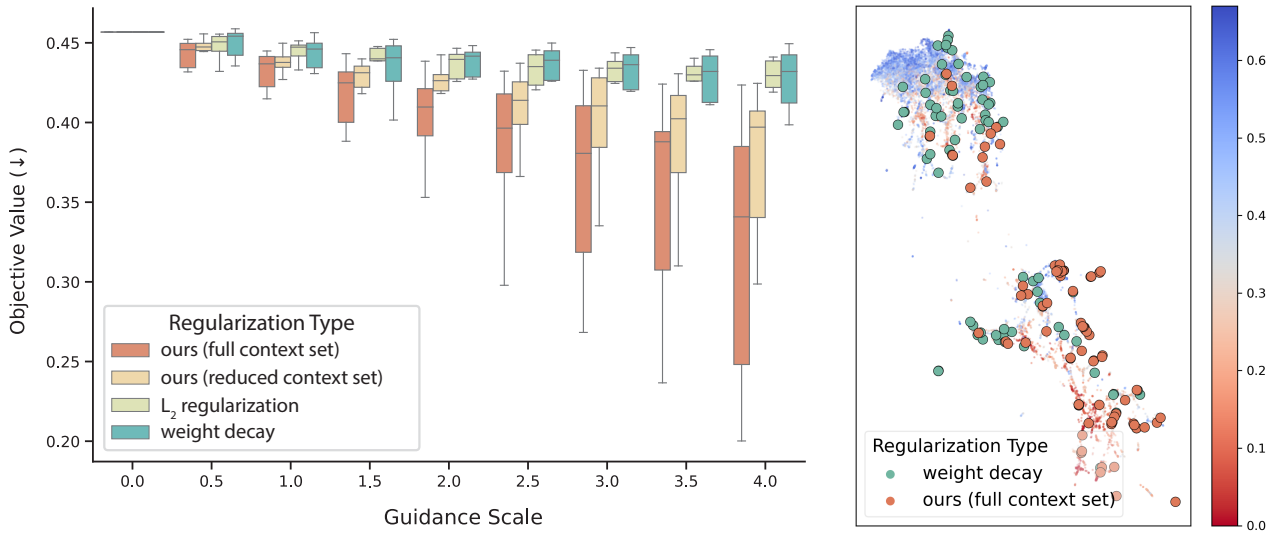


Figure 5: Comparison of polycyclic aromatic systems generated with different guidance models across ten independent training and sampling runs. **Left:** Full distribution of generated objective values (↓) ablated over different context sets and guidance scales. **Right:** UMAP plot (McInnes et al., 2018) of training (upper left) and test set (lower right), as well as samples from guided diffusion models. Validity and novelty are analyzed in Appendix B.3 and show similar trends.

5.2. Equivariant Diffusion For Materials

We additionally evaluate our method by applying it to equivariant diffusion models for materials design. Specifically, we follow the experimental setup of Weiss et al. (2023) and train an E(3)-equivariant diffusion model (Hoogeboom et al., 2022) on all polycyclic aromatic systems consisting of up to 11 cata-condensed benzene rings (Wahab et al., 2022). As in Weiss et al. (2023), we guide the sampling process towards molecules with desirable electronic properties by leveraging the gradients of an E(3)-equivariant graph neural network (Satorras et al., 2021) to minimize a composite objective of adiabatic ionization potential, electron affinity, and HOMO-LUMO gap.

In contrast to the experimental setting in Section 5.1, this application considers an exhaustively enumerated search space, allowing us to construct clustered data splits in which maximal generalization is required to reach held-out regions with optimal objective values. Using this setup, we train a diffusion model and different guidance networks on a low-value training set and examine how well they are able to recover molecules from held-out regions with desirable electronic properties. We present a comparison of different methods in Figure 5, confirming that our approach performs significantly better than standard regularization techniques across guidance scales. We additionally ablate the information content of the context set by removing the most relevant entries, observing that these are a key determinant of our method's success. Full experimental details and additional results are provided in Appendix B.3.

5.3. Discrete Diffusion for Protein Sequences

Furthermore, we explore how well our method extends to categorical diffusion models in the context of protein sequence optimization. In particular, we follow the experimental framework of Gruver et al. (2023) to apply context-guided diffusion to the property-conditioned infilling of antibody complementarity-determining regions. In contrast to the applications in Sections 5.1 and 5.2, the guidance model of Gruver et al. (2023) is not a separate network, but rather a regression head of the score model.Specifically, a masked language model (Austin et al., 2021; Bhargava et al., 2021) is used to learn sequence embeddings from which a linear output head and a feed-forward network estimate the scores and properties, respectively.

At sampling time, Gruver et al. (2023) generate sequences with a grid of denoising hyperparameters and examine the resulting Pareto front of objective values versus "naturalness" i.e. how likely a sequence is to be synthesizable, estimated by its likelihood under a protein language model (Ferruz et al., 2022). Following this approach, we train different guidance models and use them to generate samples with the same hyperparameter grid as Gruver et al. (2023). The corresponding Pareto fronts are presented in Figure 6. While standard methods fare well in in-distribution settings, we observe that models trained with our regularizer outperform them as samples progress into an out-of-distribution regime, consistently producing better properties for a given level of naturalness. Full experimental details and additional results are provided in Appendix B.4.

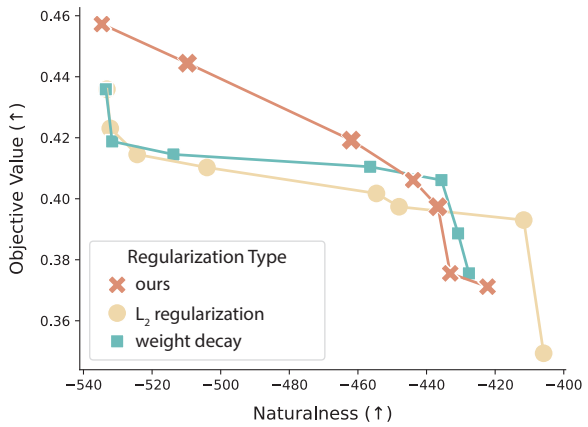


Figure 6: Pareto fronts of samples generated with different regularization schemes, highlighting the trade-off between objective value (↑) and naturalness (↑). As samples move away from the training data and enter an out-of-distribution regime, our method consistently generates sequences with better properties for any given level of naturalness.

6. Discussion and Limitations

We conducted a comprehensive empirical evaluation of our method across three distinct molecular and protein design tasks, observing consistent and significant improvements over existing state-of-the-art guided diffusion models. By leveraging unlabeled data to train guidance functions that exhibit high uncertainty and well-behaved gradients on out-of-distribution context points, our context-guided models enabled the more reliable generation of novel molecules with improved properties. These gains were most pronounced when generalization from scarce and biased training data was required to generate compounds from unseen, high-value subsets of chemical and protein sequence space. Our results highlight that even relatively simple regularization techniques can yield substantial empirical benefits when combined with relevant context sets.

The proposed method is best suited for domains in which unlabeled data is abundant or can be easily generated. While it does not incur any computational overhead at sampling time, where hundreds of model evaluations are required to generate a single molecule, it does come with an increased cost during training. Specifically, computing the regularization term requires two additional forward passes per training iteration to generate the context set embeddings and predictions. Fortunately, we found this added cost to be moderate in absolute terms (see Appendix B.5), as guidance models tend to be relatively lightweight. Additionally, we provide a sensitivity study over context batch sizes in Table 1, which suggests that the computational overhead can be further reduced while maintaining favorable empirical performance.

Table 1: Ablation over the context batch size used by our regularizer. Larger context batches generally yield better results but incur higher computational costs. We find that the context batch size can be reduced substantially with only a marginal deterioration of model performance. Performance metrics and runtimes are averaged across 5 independent training and sampling runs with different random seeds.

Graph-Structured Diffusion for Small Molecules (Section 5.1)

Batch Size	Batch Time (s)	PARP1 (↑)	FA7 (↑)	5HT1B (↑)
128	0.09±0.01	**0.66±0.01**	**0.48±0.01**	**0.57±0.01**
64	0.08±0.00	**0.66±0.01**	**0.48±0.01**	**0.57±0.01**
32	**0.07±0.00**	**0.66±0.01**	**0.48±0.01**	**0.57±0.02**
16	**0.07±0.00**	0.65±0.02	0.47±0.02	0.56±0.03
8	**0.07±0.00**	0.62±0.03	0.45±0.03	0.53±0.03

Equivariant Diffusion For Materials (Section 5.2)

Batch Size	Batch Time (s)	scale=0 (↓)	scale=2 (↓)	scale=4 (↓)
128	0.28±0.02	0.45±0.00	**0.39±0.01**	**0.34±0.01**
64	0.25±0.03	0.45±0.00	**0.40±0.01**	**0.34±0.02**
32	0.26±0.01	0.45±0.00	0.42±0.01	0.36±0.01
16	0.24±0.02	0.45±0.00	0.41±0.01	0.38±0.03
8	**0.23±0.01**	0.45±0.00	0.42±0.01	0.39±0.02

Beyond choosing the context batch size, our approach requires the optimization of two additional hyperparameters: the covariance scale σ_t and the diagonal offset parameter τ_t. Additionally, we find that the choice of the context set significantly impacts model performance, requiring sufficient domain expertise to ensure the inclusion of informative and relevant unlabeled context points.

7. Conclusions

In this paper, we introduced context-guided diffusion, a simple plug-and-play method that leverages unlabeled data and smoothness constraints to improve the out-of-distribution generalization of guided diffusion models. We demonstrated the versatility of this approach by applying it to the design of small molecules, materials, and proteins, achieving substantial performance gains in each domain. Several promising directions for future work exist, such as encoding more complex out-of-distribution behavior into the guidance model, for instance by reverting to the outputs of physics-based methods rather than high predictive uncertainty. Another promising avenue is the construction of maximally informative context sets, potentially through active learning strategies that iteratively select unlabeled data points based on their relevance and potential to improve guidance model performance. Exploring the integration of context-guided diffusion with techniques such as multi-task learning or meta-learning could also enable the development of more versatile guidance functions that leverage knowledge from related domains or tasks. We believe that exploring this line of research has the potential to lead to more adaptable and robust diffusion models that are able to solve a wider range of challenging real-world problems.

Impact Statement

This work introduces a regularization technique to improve the performance of guided diffusion models for inverse molecular design. These methods have the potential to accelerate the discovery and development of new medicines, functional materials, and sustainable alternatives that could benefit society (Wang et al., 2023). However, like any powerful technology, there is also the risk of dual-use or misuse, for example for the generation of hazardous agents that could endanger public safety (Urbina et al., 2022).

Acknowledgements

We thank the anonymous reviewers for their useful feedback. LK acknowledges financial support from the University of Oxford's Clarendon Fund and F. Hoffmann-La Roche AG. We gratefully acknowledge the Oxford Advanced Research Computing service for providing computing resources and infrastructure.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Alhossary, A., Handoko, S. D., Mu, Y., and Kwoh, C.-K. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Baek, J., Kang, M., and Hwang, S. J. Accurate learning of graph representations with graph multiset pooling. *arXiv preprint arXiv:2102.11533*, 2021.

Bajusz, D., Rácz, A., and Héberger, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):1–13, 2015.

Band, N., Rudner, T. G. J., Feng, Q., Filos, A., Nado, Z., Dusenberry, M. W., Jerfel, G., Tran, D., and Gal, Y. Benchmarking Bayesian deep learning on diabetic retinopathy detection tasks. In *Advances in Neural Information Processing Systems 34*, 2021.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

Bhargava, P., Drozd, A., and Rogers, A. Generalization in nli: Ways (not) to go beyond simple heuristics. *arXiv preprint arXiv:2110.01518*, 2021.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.

Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., and Jensen, K. F. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1608, 2022.

Bohacek, R. S., McMartin, C., and Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal Research Reviews*, 16 (1):3–50, 1996.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. S. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kKF8_K-mBbS.

De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Dunbar, J. and Deane, C. M. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 2016.

Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C. M. Sabdab: the structural antibody database. *Nucleic Acids Research*, 42 (D1):D1140–D1146, 2014.

Ertl, P. Cheminformatics analysis of organic substituents: identification of the most common substituents, calculation of substituent properties, and automatic identification of drug-like bioisosteric groups. *Journal of Chemical Information and Computer Sciences*, 43(2):374–380, 2003.

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1:1–11, 2009.

Ferruz, N., Schmidt, S., and Höcker, B. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

Fishman, N., Klarner, L., Bortoli, V. D., Mathieu, E., and Hutchinson, M. J. Diffusion models for constrained domains. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL https://openreview.net/forum?id=xuWTFQ4VGO. Expert Certification.

Fishman, N., Klarner, L., Mathieu, E., Hutchinson, M. J., and Bortoli, V. D. Metropolis sampling for constrained diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=jzseUq55eP.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gruver, N., Stanton, S., Frey, N. C., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. Protein design with guided discrete diffusion. *arXiv preprint arXiv:2305.20009*, 2023.

Guan, J., Zhou, X., Yang, Y., Bao, Y., Peng, J., Ma, J., Liu, Q., Wang, L., and Gu, Q. DecompDiff: Diffusion models with decomposed priors for structure-based drug design. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11827–11846. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/guan23a.html.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Haussmann, U. G. and Pardoux, E. Time reversal of diffusions. *The Annals of Probability*, pp. 1188–1205, 1986.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Huang, C.-W., Aghajohari, M., Bose, J., Panangaden, P., and Courville, A. C. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35: 2750–2761, 2022.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., et al. Illuminating protein space with a programmable generative model. *Nature*, pp. 1–9, 2023.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.

Isert, C., Atz, K., Jiménez-Luna, J., and Schneider, G. Qmugs, quantum mechanical properties of drug-like molecules. *Scientific Data*, 9(1):273, 2022.

Ji, Y., Zhang, L., Wu, J., Wu, B., Li, L., Huang, L.-K., Xu, T., Rong, Y., Ren, J., Xue, D., et al. Drugood: Out-of-distribution dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8023–8031, 2023.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. S. Torsional diffusion for molecular conformer generation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=w6fj2r62r_H.

Jo, J., Lee, S., and Hwang, S. J. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pp. 10362–10383. PMLR, 2022.

Kadkhodaie, Z., Guth, F., Simoncelli, E. P., and Mallat, S. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024.

Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4893–4902, 2019.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Klarner, L., Rudner, T. G., Reutlinger, M., Schindler, T., Morris, G. M., Deane, C., and Teh, Y. W. Drug discovery under covariate shift with domain-informed prior distributions over functions. In *International Conference on Machine Learning*, pp. 17176–17197. PMLR, 2023.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

Landrum, G. et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.

Lee, S., Jo, J., and Hwang, S. J. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pp. 18872–18892. PMLR, 2023.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9641–9650, 2020.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512, 2020.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lou, A. and Ermon, S. Reflected diffusion models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22675–22701. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lou23a.html.

Marsland, S. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.

Maynard Smith, J. Natural selection and the concept of a protein space. *Nature*, 225(5232):563–564, 1970.

McInnes, L., Healy, J., and Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Mysinger, M. M., Carchia, M., Irwin, J. J., and Shoichet, B. K. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of Medicinal Chemistry*, 55(14):6582–6594, 2012.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814, 2010.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Olsen, T. H., Boyles, F., and Deane, C. M. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

pandas development team, T. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Peng, X., Guan, J., Liu, Q., and Ma, J. MolDiff: Addressing the atom-bond inconsistency problem in 3D molecule diffusion generation. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27611–27629. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/peng23b.html.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50 (5):742–754, 2010.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Rudner, T. G., Chen, Z., Teh, Y. W., and Gal, Y. Tractable function-space variational inference in Bayesian neural networks. *Advances in Neural Information Processing Systems*, 35:22686–22698, 2022.

Rudner, T. G., Kapoor, S., Qiu, S., and Wilson, A. G. Function-space regularization in neural networks: A probabilistic perspective. In *International Conference on Machine Learning*, pp. 29275–29290. PMLR, 2023.

Rudner, T. G. J., Lu, C., Osborne, M. A., Gal, Y., and Teh, Y. W. On pathologies in KL-regularized reinforcement learning from expert demonstrations. In *Advances in Neural Information Processing Systems 34*, 2021.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Sanchez-Lengeling, B. and Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pp. 9323–9332. PMLR, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Stiefl, N., Watson, I. A., Baumann, K., and Zaliani, A. Erg: 2d pharmacophore descriptions for scaffold hopping. *Journal of Chemical Information and Modeling*, 46(1): 208–220, 2006.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tingle, B. I., Tang, K. G., Castanon, M., Gutierrez, J. J., Khurelbaatar, M., Dandarchuluun, C., Moroz, Y. S., and Irwin, J. J. ZINC-22 - a free multi-billion-scale database of tangible compounds for ligand discovery. *Journal of Chemical Information and Modeling*, 63(4):1166–1176, 2023.

Tran, D., Liu, J., Dusenberry, M. W., Phan, D., Collier, M., Ren, J., Han, K., Wang, Z., Mariet, Z., Hu, H., Band, N., Rudner, T. G. J., Singhal, K., Nado, Z., van Amersfoort, J., Kirsch, A., Jenatton, R., Thain, N., Yuan, H., Buchanan, K., Murphy, K., Sculley, D., Gal, Y., Ghahramani, Z., Snoek, J., and Lakshminarayanan, B. Plex: Towards reliability using pretrained large model extensions. In *ICML Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward*, 2022.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4068–4076, 2015.

Urbina, F., Lentzos, F., Invernizzi, C., and Ekins, S. Dual use of artificial-intelligence-powered drug discovery. *Nature Machine Intelligence*, 4(3):189–191, 2022.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pp. 9690–9700. PMLR, 2020.

Van Rossum, G. and Drake, F. L. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Wahab, A., Pfuderer, L., Paenurk, E., and Gershoni-Poranne, R. The compas project: A computational database of polycyclic aromatic systems. phase 1: cata-condensed polybenzenoid hydrocarbons. *Journal of Chemical Information and Modeling*, 62(16):3704–3713, 2022.

Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., Chandak, P., Liu, S., Van Katwyk, P., Deac, A., et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

Waskom, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Weiss, T., Mayo Yanes, E., Chakraborty, S., Cosmo, L., Bronstein, A. M., and Gershoni-Poranne, R. Guided diffusion for inverse molecular design. *Nature Computational Science*, 3(10):873–882, 2023.

Xu, M., Powers, A. S., Dror, R. O., Ermon, S., and Leskovec, J. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pp. 38592–38610. PMLR, 2023.

Zaidi, S., Schaarschmidt, M., Martens, J., Kim, H., Teh, Y. W., Sanchez-Gonzalez, A., Battaglia, P., Pascanu, R., and Godwin, J. Pre-training via denoising for molecular property prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Appendix

Table of Contents

A Guidance Function Parameterization **17**

A.1 Probabilistic Interpretation 17

B Experimental Details **18**

B.1 Illustrating the Behaviour of Guided Diffusion Models on the Swiss Roll Dataset 18

B.2 Graph-Structured Diffusion for Small Molecules 21

B.3 Equivariant Diffusion for Materials. 31

B.4 Discrete Diffusion for Protein Sequences. 35

B.5 Runtime Comparison. 37

B.6 Similarity of the Generated Samples to the Training and Validation Sets. 38

Reproducibility

All data and code needed to reproduce our experiments can be accessed at:

`https://github.com/leojklarner/context-guided-diffusion`

A. Guidance Function Parameterization

To guide sampling in a diffusion model toward certain desired properties, a *guidance model* $f_t(\cdot;\theta)$ is trained on a dataset $\mathcal{D}$, a parametric *guidance function* $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ is defined using the learned guidance model, and then the *conditional gradient/score function*

$$\nabla \log p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta)) + \nabla \log p(\mathbf{x}_t), \tag{A.1}$$

are used to guide generation towards high-density regions $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$. To provide a general framework that accommodates both classification and regression settings, we let $\mathbf{Y}$ denote an "optimality" random variable indicating the optimality (under some domain-specific optimality function) of a given input $\mathbf{x}_t$.

For discrete desirable properties, the guidance model is trained on a classification task, and the guidance function $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ is defined as a categorical likelihood function, making $-\log p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ a cross-entropy loss. To guide a diffusion model towards generating samples consistent with a specific class c, the guidance function is computed under $\mathbf{y} = \text{one-hot}(c)$, denoting that the desired class c is represented by a one-hot encoding. The score function then becomes the class probability of the desired class under the learned model:

$$\nabla \log p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta)) = \nabla \log \text{softmax}(f_t(\mathbf{x} \mid \theta))_c. \tag{A.2}$$

For continuous desirable properties, the guidance model is trained on a regression task, and the guidance function $p(\mathbf{y} \mid \mathbf{x}_t; f_t(\cdot;\theta))$ is defined as a Bernoulli likelihood function with

$$p(\mathbf{y} = 1 \,|\, \mathbf{x}_t, C; f_t(\cdot;\theta)) = \exp\left(f_t(\mathbf{x}_t;\theta) - C\right), \tag{A.3}$$

denoting the probability of optimality, and C representing a domain-specific "goal value". Importantly, this formulation requires that $f_t(\mathbf{x}_t;\theta) < C$ so that increases in $f_t(\mathbf{x}_t;\theta))$ indicate desirable properties.

A.1. Probabilistic Interpretation

Letting $\mathcal{O}$ be an "optimality" random variable indicating whether a prediction $f_t(\hat{\mathbf{x}}_t;\theta)$ is optimal for achieving a desired behavior (e.g., high predictive uncertainty on data points far away from the training data), we can define a Bernoulli observation model where

$$p(\mathcal{O} = 1 \,|\, \theta, t; f_t, p_{\hat{\mathbf{X}}_t}) = \exp\left(-R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})\right), \tag{A.4}$$

denotes the probability of optimality as a function of the regularizer. That is, the smaller $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$, the higher the probability that a prediction $f_t(\hat{\mathbf{x}}_t;\theta)$ is optimal.

Following the result in Rudner et al. (2023), we note that specifying a data-dependent regularizer of the form shown in Equation (8) corresponds to specifying a data-driven prior distribution over neural network parameters

$$p(\theta \,|\, t, \mathcal{O} = 1; f_t, p_{\hat{\mathbf{X}}_t}) = \frac{p(\mathcal{O} = 1 \,|\, \theta, t; f_t, p_{\hat{\mathbf{X}}_t})\, p(\theta)}{p(\mathcal{O} = 1 \,|\, t; f_t, p_{\hat{\mathbf{X}}_t})}.$$

Minimizing the objective function in Equation (9) corresponds to finding a variational approximation to the posterior implied by this data-driven prior and the observed data. That is, minimizing the objective function in Equation (9) (approximately) corresponds to finding a variational distribution $q_\Theta = \mathcal{N}(\theta; \theta', \sigma^2)$ (with fixed and very small σ^2) that solves the variational problem

$$\min_{q_\Theta} \mathbb{E}_{p_\mathcal{T}} \left[\mathbb{D}_{\text{KL}}(q_\Theta \,||\, p_{\Theta|\mathcal{D}_\mathcal{T},\mathcal{O},\mathcal{T}})\right], \tag{A.5}$$

allowing us to interpret the learned guidance model through the lens of Bayesian inference.

B. Experimental Details

This section provides additional information and experimental results complementing the main text. Appendix B.1 details the regression experiments from Section 3 and contrasts our approach with standard L_2 regularization, illustrated in Figure 3. Appendix B.2 presents our extension of Lee et al. (2023)'s work on graph-structured diffusion models for small molecule generation (Section 5.1). Appendix B.3 presents our extension of Weiss et al. (2023)'s experiments on equivariant diffusion models for generating novel polycyclic aromatic systems (Section 5.2). Finally, Appendix B.4 presents our extension of Gruver et al. (2023)'s approach to optimizing discrete protein sequences using categorical diffusion models (Section 5.3).

All code was written in (Van Rossum & Drake, 2009) and can be accessed at https://github.com/leojklarner/context-guided-diffusion. A range of core scientific computing libraries were used for data preparation and analysis, including NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Pandas (pandas development team, 2020), Matplotlib (Hunter, 2007), Seaborn (Waskom, 2021), Scikit-learn (Pedregosa et al., 2011) and RDKit (Landrum et al., 2013). All deep learning models were implemented in PyTorch (Paszke et al., 2019).

B.1. Illustrating the Behaviour of Guided Diffusion Models on the Swiss Roll Dataset

Dataset. In order to illustrate the impact of our regularizer on the learning dynamics of a guidance model, particularly when only trained on a constrained and less diverse subset of the input domain, we conduct a comparison with standard parameter-space regularization schemes, namely explicit L_2 regularization. This comparison is performed on the Swiss roll dataset, as described by Marsland (2011) and implemented in the Scikit-learn library (Pedregosa et al., 2011). The input data $\mathbf{X}_t \in \mathbb{R}^2$ and regression labels $\mathbf{y} \in \mathbb{R}$ are depicted in Figure 7. A threshold of $\mathbf{y} = 1$ is used to split the data into training and validation sets. The training set encompasses data points with lower label values, while the validation set consists of data points with higher labels. This split is designed to evaluate the capability of a guided diffusion model trained on a subset of lower-value data to extrapolate to and generate data points from higher-value, out-of-distribution regions.

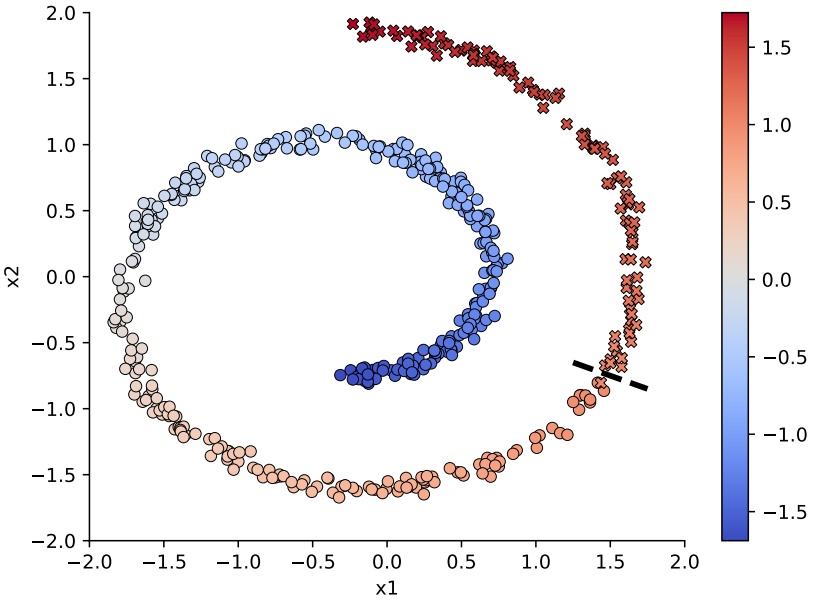


Figure 7: A visualization of the Swiss roll dataset used to train different guidance models. It consists of 500 data points generated with the corresponding function in the Scikit-learn library (Pedregosa et al., 2011) using a noise parameter of 0.3. Both covariates $\mathbf{X}_t \in \mathbb{R}^2$ and labels $\mathbf{y} \in \mathbb{R}$ have been normalized to a mean of zero and a standard deviation of one. The color-coded regression labels are derived from the data points' sequential ordering along the manifold's arc length. A label threshold of $\mathbf{y} = 1$, denoted by a dashed line, shows the training-validation split: the training data (circles) contains data points with lower label values, while the validation data (crosses) consists of data points with higher labels. This split is designed to evaluate the ability of a guided diffusion model trained on a subset of lower-value data to extrapolate to and generate data points from higher-value, out-of-distribution regions.

Diffusion Model Training. To train unconditional diffusion models, we adapt an existing implementation of the denoising diffusion probabilistic models from Ho et al. (2020) to train diffusion models on datasets in $\mathbb{R}^2$ (https://github.com/albarji/toy-diffusion). Specifically, we generate 100 000 data points from the same data-generating process as our labeled training set, i.e. data points from the Swiss roll generator with standardized labels $\mathbf{y} < 1$. Using a standard cosine beta schedule (Nichol & Dhariwal, 2021)

$$\beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, \quad \text{with} \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \quad \text{and} \quad f(t) = \cos\left(\frac{t/T + \epsilon}{1+\epsilon} \cdot \frac{\pi}{2}\right)^2 \tag{B.6}$$

where $\epsilon = 0.008$ and $T = 40$, we ensure that the noising process converges to the target distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The score network $s_\theta : \mathbb{R}^2 \times [1, \ldots, T] \to \mathbb{R}^2$ is constructed as a multi-layer perceptron with 5 hidden layers of dimension 64 and ReLU activation functions (Nair & Hinton, 2010). It is trained with the L_{simple} noise prediction objective from Ho et al. (2020) over 100 epochs of stochastic gradient descent. Specifically, we use the ADAM optimizer (Kingma & Ba, 2014) with a batch size of 2048 and a learning rate of 1×10^{-3} that drops off to 1×10^{-5} following a linear learning rate decay schedule. Samples from this unconditional model are shown in Figure 8.

Guidance Function Training. For the training of guidance models $f_t(\cdot; \theta) : \mathbb{R}^2 \times [1, \ldots, T] \to \mathbb{R}^2$, we use smaller multi-layer perceptrons with 3 hidden layers of dimension 32, applying $\sin$ activation functions and dropout (Srivastava et al., 2014) with $p = 0.2$ after each layer. These networks are trained to output both the most likely regression label $\boldsymbol{\mu}_t(\mathbf{x}_t; \theta) = f_t^1(\mathbf{x}_t; \theta)$, as well as log-variances $\log \boldsymbol{\sigma}_t^2(\mathbf{x}_t; \theta) = f_t^2(\mathbf{x}_t; \theta)$ that serve as an estimator of their predictive uncertainty. Given the regression labels $\mathbf{y}$, the models are optimized with respect to a negative log-likelihood loss $\mathcal{L}_t = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \mathbf{I})$ over 100 epochs of stochastic gradient descent, using the ADAM optimizer with a batch size of 128 and a learning rate of 1×10^{-2}. The corresponding objective $\mathcal{L}$ with an explicit L_2 regularization term is given by

$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}) \doteq \mathbb{E}_{p_\mathcal{T}} \left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\mathcal{T}, \boldsymbol{\sigma}_\mathcal{T}^2 \mathbf{I}) \right] + \frac{1}{2\lambda} ||\theta||_2^2, \tag{B.7}$$

where the regularization strength λ is optimized as a hyperparameter. Similarly, we train context-guided models by adding the additional regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ as introduced in Section 3 and given by

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}} \left[\sum_{j=1}^{2} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right)^\top K_t(\hat{\mathbf{x}}_t)^{-1} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right) \right], \tag{B.8}$$

Here, $\hat{\mathbf{x}}_t \sim p_{\hat{\mathbf{X}}_t}(\hat{\mathbf{x}}_t)$ represents a context batch sampled from an out-of-distribution context set comprising 10 000 points distributed uniformly over $[-2.5, 2.5]^2 \subset \mathbb{R}^2$. As detailed in Section 3, we use a randomly initialized parameter set ϕ to generate 32-dimensional context set embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M \times 32}$ and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{x}}_t) = \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}.$$

As the model returns mean and log-variance estimates $f_t^1(\hat{\mathbf{x}}_t; \theta)$ and $f_t^2(\hat{\mathbf{x}}_t; \theta)$, the target function values $m_t(\hat{\mathbf{x}}_t)$ are given by the training set mean $m_t^1(\hat{\mathbf{x}}_t) = -0.38$ and a log-variance hyperparameter $m_t^2(\hat{\mathbf{x}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 0.7$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx 2$ in out-of-distribution regions of the input domain. The values of the covariance scale σ_t and the diagonal offset τ_t are optimized as hyperparameters. While empirical performance tends to improve when σ_t and τ_t are made time-dependent, we found that in this setting we can already achieve good performance when keeping them constant. The resulting augmented training objective is given as

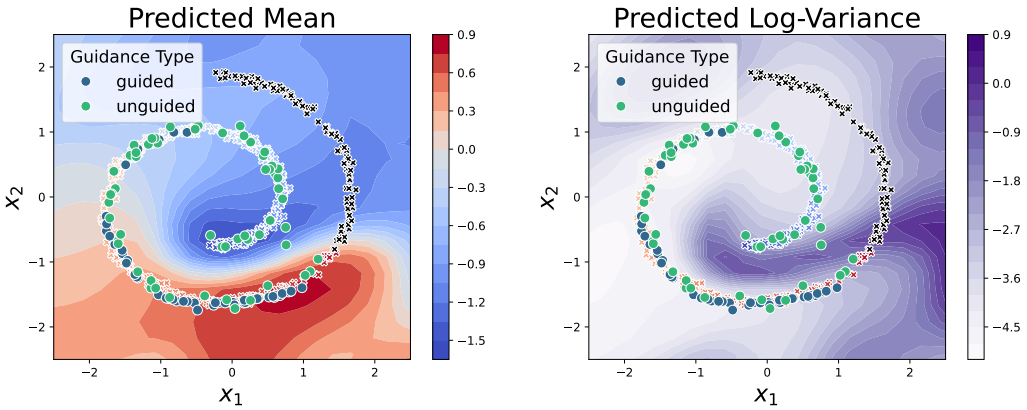
$$\mathcal{L}^*(\theta, \mathcal{D}) = \mathbb{E}_{p_\mathcal{T}} \left[\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T}) + R(\theta, f_t, \mathcal{T}, p_{\hat{\mathbf{X}}_\mathcal{T}}) \right]. \tag{B.9}$$

The optimal regularization hyperparameters were selected by performing grid search over the hyperparameter grid presented in Table 2 with respect to the supervised loss $\mathcal{L}$ evaluated on the held-out validation set.

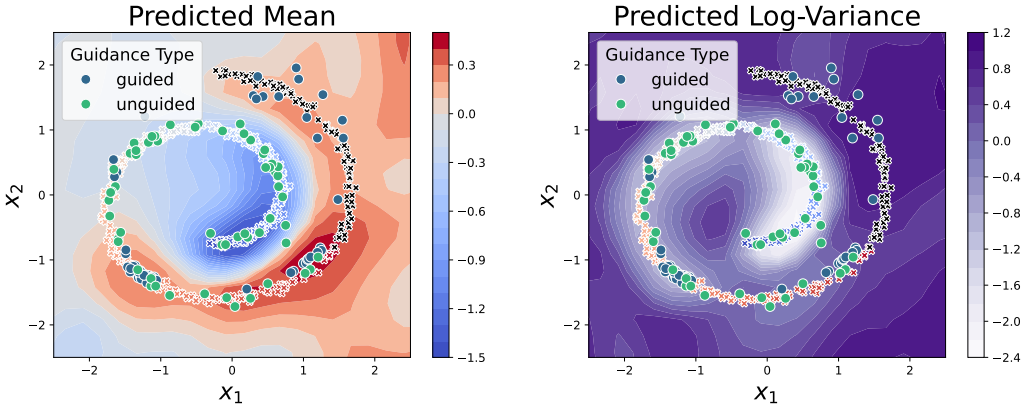
Table 2: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.1.

Hyperparameter	Description	Search Space
λ	L_2 regularization strength	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4
σ_t	covariance scale	10^{-5}, 10^{-3}, 10^{-1}, 1, 10^1, 10^3, 10^5
τ_t	diagonal offset	10^{-5}, 10^{-3}, 10^{-1}, 1, 10^1, 10^3, 10^5

Results A comparative analysis of model performances is provided in Figure 8. We observe that both L_2-regularized and context-guided models fit the training data well. However, without additional information about the input domain, the predictions of the L_2-regularized model exhibit poor generalization and miscalibrated confidence estimates when evaluated in new, unseen regions. In contrast, the context-guded regularizer is able to use this additional information to steer guidance model training toward functions that seem to capture the underlying structure of the input domain more accurately. More importantly, it also generates significantly better-calibrated uncertainty estimates. This improvement translates into higher-quality samples when using gradients from the context-guided model for conditional sampling.



(a) Predictions and samples from a context-guided model



(b) Predictions and samples from a context-guided guidance model.

Figure 8: Comparative analysis of predictions, uncertainty estimates, and samples generated by guidance models trained with L_2 regularization and context-guidance schemes. The training and validation sets, (see Figure 7), are depicted by color-coded and black crosses, respectively. We display 50 samples generated by both unconditional and conditional models, depicted as green and blue circles, respectively. The samples are generated with the same random seed and no cherry-picking. The left panel shows the model's mean predictions across a broader input space $[-2.5, 2.5]^2 \subset \mathbb{R}^2$, with warmer colors (red) indicating higher predicted values. The right panel shows the model's log-variance estimates across the same interval, serving as a measure of uncertainty, with darker regions indicating more uncertain predictons. We observe that both L_2-regularized and context-guided models fit the training data well. However, without additional information about the input domain, the predictions of the L_2-regularized model exhibit poor generalization and miscalibrated confidence estimates when evaluated in new, unseen regions. In contrast, our context-guided approach is able to use this additional information to steer guidance model training toward functions that seem to capture the underlying structure of the input domain more accurately. More importantly, it also generates significantly better-calibrated uncertainty estimates. This improvement translates into higher-quality samples when using gradients from the context-guided model for conditional sampling.

B.2. Graph-Structured Diffusion for Small Molecules

Dataset. We evaluate all methods on the same dataset as Lee et al. (2023), consisting of 250 000 molecules sampled uniformly from the ZINC database of commercially available compounds (Irwin et al., 2012). Each molecule is annotated with ground-truth labels $\mathbf{y}$, defined in terms of

- the clipped and normalized negative QuickVina docking score $\mathbf{y}_{\text{vina}}$ (Alhossary et al., 2015),
- the normalized synthetic accessibility (SA) score $\mathbf{y}_{\text{SA}}$ (Ertl, 2003), and
- the quantitative estimate for drug-likeness (QED) score $\mathbf{y}_{\text{QED}}$ (Bickerton et al., 2012).

The resulting composite objective (for which larger values indicate better performance) is given by

$$\mathbf{y} = \frac{\text{clip}(-\mathbf{y}_{\text{vina}}, 20, 0)}{20} \cdot \frac{10 - \mathbf{y}_{\text{SA}}}{9} \cdot \mathbf{y}_{\text{QED}}\,, \text{ with } \mathbf{y} \in [0, 1]. \tag{B.10}$$

As the QuickVina docking score $\mathbf{y}_{\text{vina}}$ is a function of the specific protein pocket that a molecule is docked into, all data preparation and model training steps described in this section are repeated across five different proteins from the DUD-E dataset (Mysinger et al., 2012) selected by Lee et al. (2023), namely PARP1 (Poly [ADP-ribose] polymerase-1), FA7 (Coagulation factor VII), 5HT1B (5-hydroxytryptamine receptor 1B), BRAF (Serine/threonine-protein kinase B-raf), and JAK2 (Tyrosine-protein kinase JAK2). All compounds are featurized as topological molecular graphs $\mathbf{g}_0$, defined as tuples $\mathbf{g}_0 = (\mathbf{x}_0, \mathbf{a}_0)$ of node features $\mathbf{x}_0 \in \mathbb{R}^{M \times F}$ and adjacency matrices $\mathbf{a}_0 \in \mathbb{R}^{M \times M}$, where $M = 38$ is the maximum number of heavy atoms in the training set and $F = 10$ is the dimensionality of node features, denoting a one-hot encoding of the element types C, N, O, F, P, S, Cl, Br, I, and NONE. Different bond types are learned by discretizing the continuous adjacency matric $\mathbf{a}_0$ into bins corresponding to certain bonds, namely $a_{i,j} < 0.5 \rightarrow$ NONE, $0.5 < a_{i,j} < 1.5 \rightarrow$ single, $1.5 < a_{i,j} < 2.5 \rightarrow$ double, $a_{i,j} > 2.5 \rightarrow$ triple. Following Lee et al. (2023), we then order the molecules by their labels $\mathbf{y}$ and split them evenly into a low-value training set and a high-value test set of 125 000 data points each, repeated for each of the five protein targets. This experimental setup allows us to evaluate the ability of guided diffusion models trained on molecules with low-value properties to extrapolate to and generate data points from higher-value, out-of-distribution regions.

To generate a context set $\hat{\mathbf{x}}_{\mathcal{C}}$ for our context-guided diffusion method, an additional 500 000 unlabeled compounds were sampled uniformly from the ZINC database and processed in the same fashion as before. A comparison of the training and context set in terms of different molecular properties derived with RDKIT (Landrum et al., 2013) is presented in Figure 9.

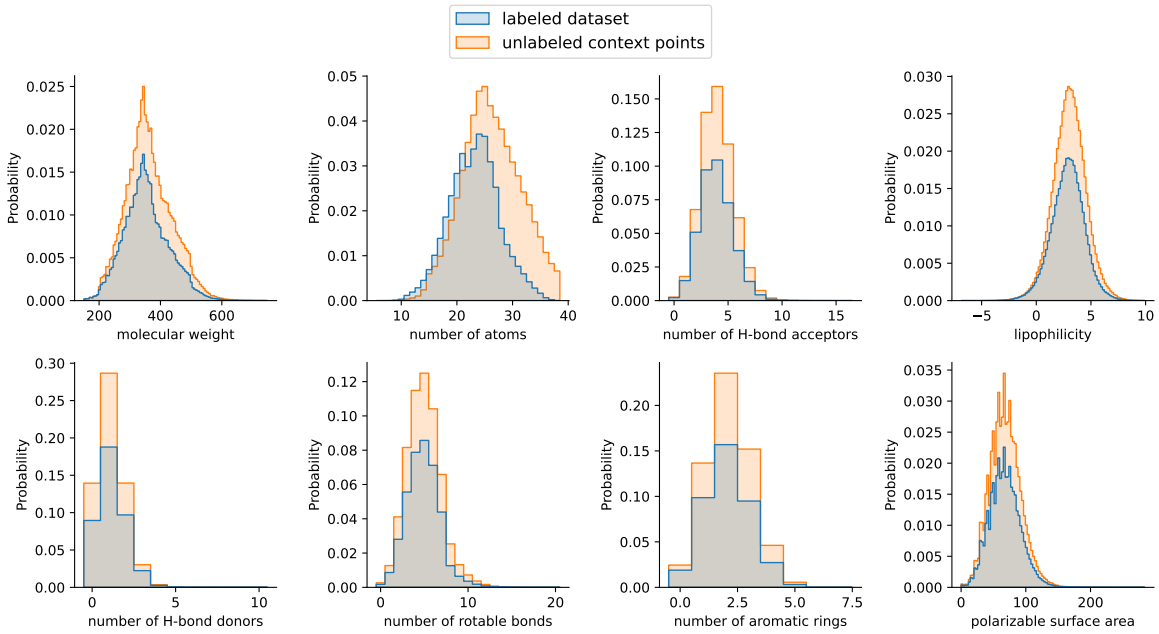


Figure 9: Comparison of the molecular properties of the labeled training and unlabeled context sets, verifying that the latter consists of meaningfully related yet distinct compounds and is suitable for use with our regularization scheme.

Diffusion Models over Graphs. The *Molecular Out-Of-distribution Diffusion* (MOOD) framework introduced in Lee et al. (2023) leverages diffusion models defined over the space of molecular graphs. Specifically, they employ the *Graph Diffusion via the System of Stochastic differential equations* (GDSS) formalism of Jo et al. (2022), which we restate in the following. In this framework, a molecular graph $\mathbf{g}_0$ is defined as a tuple $\mathbf{g}_0 = (\mathbf{x}_0, \mathbf{a}_0)$ of node features $\mathbf{x}_0 \in \mathbb{R}^{M \times F}$ and adjacency matrices $\mathbf{a}_0 \in \mathbb{R}^{M \times M}$, where M is the maximum number of heavy atoms in the training set and F is the dimensionality of the node features. Following the formalism of Song et al. (2020) outlined in Section 2, the forward diffusion process $\{\mathbf{G}_t = (\mathbf{X}_t, \mathbf{A}_t)\}_{t=0}^{T}$ is given by

$$\mathrm{d}\mathbf{G}_t = \mathbf{f}(\mathbf{G}_t, t)\,\mathrm{d}t + \beta_t\,\mathrm{d}\mathbf{B}_t \tag{B.11}$$

and decomposes into a system of stochastic differential equations that can be modeled jointly

$$\begin{aligned} \mathrm{d}\mathbf{X}_t &= \mathbf{f}_{\mathbf{X}}(\mathbf{X}_t, t)\,\mathrm{d}t + \beta_{\mathbf{X},t}\,\mathrm{d}\mathbf{B}_t \\ \mathrm{d}\mathbf{A}_t &= \mathbf{f}_{\mathbf{A}}(\mathbf{A}_t, t)\,\mathrm{d}t + \beta_{\mathbf{A},t}\,\mathrm{d}\mathbf{B}_t, \end{aligned} \tag{B.12}$$

where $\mathbf{f}(\mathbf{G}_t, t) = [\mathbf{f}_{\mathbf{X}}(\mathbf{X}_t, t), \mathbf{f}_{\mathbf{A}}(\mathbf{A}_t, t)]$ and $\beta_t = [\beta_{\mathbf{X},t}, \beta_{\mathbf{A},t}]$. The corresponding time reversal $\{\bar{\mathbf{G}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t)\}_{t \geq 0} = \{\mathbf{G}_{T-t} = (\mathbf{X}_{T-t}, \mathbf{A}_{T-t})\}_{t \in [0,T]}$ is given by

$$\mathrm{d}\bar{\mathbf{G}}_t = \left\{-\mathbf{f}(\bar{\mathbf{G}}_t, T-t) + \beta_{T-t}^2 \nabla_{\mathbf{G}} \log p_{T-t}\left(\bar{\mathbf{G}}_t\right)\right\}\,\mathrm{d}t + \beta_{T-t}\,\mathrm{d}\mathbf{B}_t, \tag{B.13}$$

flowing backward in time and decomposing into a corresponding joint system of SDEs given by

$$\begin{aligned} \mathrm{d}\bar{\mathbf{X}}_t &= \left\{-\mathbf{f}_{\mathbf{X}}(\bar{\mathbf{X}}_t, T-t) + \beta_{\mathbf{X},T-t}^2 \nabla_{\mathbf{X}} \log p_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t\right)\right\}\,\mathrm{d}t + \beta_{\mathbf{X},T-t}\,\mathrm{d}\mathbf{B}_t \\ \mathrm{d}\bar{\mathbf{A}}_t &= \left\{-\mathbf{f}_{\mathbf{A}}(\bar{\mathbf{A}}_t, T-t) + \beta_{\mathbf{A},T-t}^2 \nabla_{\mathbf{A}} \log p_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t\right)\right\}\,\mathrm{d}t + \beta_{\mathbf{A},T-t}\,\mathrm{d}\mathbf{B}_t, \end{aligned} \tag{B.14}$$

where $\nabla_{\mathbf{X}} \log p_t(\cdot)$ and $\nabla_{\mathbf{A}} \log p_t(\cdot)$ that are approximated by separate score networks $s_{\theta_{\mathbf{X}}}(\mathbf{X}_t, \mathbf{A}_t, t)$ and $s_{\theta_{\mathbf{A}}}(\mathbf{X}_t, \mathbf{A}_t, t)$. To encourage the exploration of novel chemical space, Lee et al. (2023) condition the denoising process on a hyperparameter $\gamma \in [0, 1]$ that shrinks the unconditional gradients to explore near out-of-distribution regions around the training distribution

$$\begin{aligned} \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t \mid \gamma) &= \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log p_t(\gamma \mid \boldsymbol{G}_t) \\ &= \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log \left(p_t(\mathbf{G}_t)^{-\sqrt{\gamma}}\right) \\ &= (1 - \sqrt{\gamma})\,\nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t). \end{aligned} \tag{B.15}$$

In all of our experiments, we follow Lee et al. (2023) and set $\sqrt{\gamma} = 0.2$. To additionally condition the denoising process on desirable molecular properties, Lee et al. (2023) make use of standard guided diffusion as outlined in Section 2, specified as

$$\begin{aligned} \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t \mid \gamma, \mathbf{y}) &= \nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log p_t(\gamma \mid \boldsymbol{G}_t) + \nabla_{\mathbf{G}} \log p_t(\mathbf{y} \mid \mathbf{G}_t) \\ &= (1 - \sqrt{\gamma})\,\nabla_{\mathbf{G}} \log p_t(\mathbf{G}_t) + \nabla_{\mathbf{G}} \log p_t(\mathbf{y} \mid \mathbf{G}_t). \end{aligned} \tag{B.16}$$

Here, a regression model $f_t(\mathbf{g}_t, \theta)$ is used to approximate the conditional distribution $p_t(\mathbf{y} \mid \mathbf{g}_t) \approx \exp(\alpha_t f_t(\mathbf{g}_t, \theta))/Z_t$, where α_t is an adaptive guidance scale based on a hyperparameter $r_{\mathbf{G},0}$ and given by

$$\alpha_{\mathbf{G},t} = r_{\mathbf{G},t} \frac{\|s_\theta(\mathbf{G}_t, t)\|_2}{\nabla_{\mathbf{G}} \log \|\beta_\theta(\mathbf{G}_t, t)\|_2} \quad \text{with} \quad r_{\mathbf{G},t} = 0.1^t r_{\mathbf{G},0} \tag{B.17}$$

Combining the decomposition into a system of joint SDEs from Equation (B.14), the shrinking of the unconditional gradients from Equation (B.15) and the guidance model formulation from Equation (B.17), the full conditional sample generation process used by Lee et al. (2023) is given by

$$\begin{aligned} \mathrm{d}\bar{\mathbf{X}}_t &= \beta_{\mathbf{X},T-t}^2 \left\{\frac{1}{2}\bar{\mathbf{X}}_t + (1 - \sqrt{\gamma})\, s_{\theta_{\mathbf{X}}}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t, T-t\right) + \nabla_{\mathbf{X}} \log \exp\left(\alpha_{\mathbf{X},T-t} f_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t; \theta\right)\right)\right\} \mathrm{d}t + \beta_{\mathbf{X},T-t} \mathrm{d}\mathbf{B}_t \\ \mathrm{d}\bar{\mathbf{A}}_t &= \beta_{\mathbf{A},T-t}^2 \left\{(1 - \sqrt{\gamma})\, s_{\theta_{\mathbf{A}}}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t, T-t\right) + \nabla_{\mathbf{A}} \log \exp\left(\alpha_{\mathbf{A},T-t} f_{T-t}\left(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_t; \theta\right)\right)\right\} \mathrm{d}t + \beta_{\mathbf{A},T-t} \mathrm{d}\mathbf{B}_t \end{aligned} \tag{B.18}$$

where the drift and diffusion coefficients of $\mathrm{d}\bar{\mathbf{X}}_t$ and $\mathrm{d}\bar{\mathbf{A}}_t$ are determined by the variance-preserving (VP) and variance-exploding (VE) SDE from Song et al. (2020), respectively.

Diffusion Model Training. As in Lee et al. (2023), we adopt the pre-trained node feature and adjacency matrix score networks $s_{\theta_\mathbf{X}}(\mathbf{x}_t, \mathbf{a}_t, t)$ and $s_{\theta_\mathbf{A}}(\mathbf{x}_t, \mathbf{a}_t, t)$ from Jo et al. (2022) to define our unconditional diffusion model. These networks are built on graph multi-head attention (GMH) architectures (Baek et al., 2021) to ensure permutation-equivariance. Each network consists of two graph attention layers with four attention heads of hidden dimension 16 and $\tanh$ activation functions, followed by a multi-layer perceptron output head with three layers, each of dimension 82 and eLU activation functions (Clevert et al., 2015). The node feature SDE is based on the variance-preserving SDE from Song et al. (2020), using a linear β-schedule ($\beta_{\mathbf{X},0} = 0.1$, $\beta_{\mathbf{X},T} = 1$). In contrast, the adjacency matrix SDE uses the variance-exploding SDE also from Song et al. (2020), with an exponential β-schedule ($\beta_{\mathbf{A},0} = 0.2$, $\beta_{\mathbf{A},T} = 1$). Both models employ $T = 1000$ discretization steps. It's important to note that the unconditional score networks from (Jo et al., 2022) were trained on an i.i.d 85%-15% training-validation split of our dataset, meaning that they may have been optimized using part of the high-property validation set also used for training other guidance models. However, this is not an issue as (a) the unconditional diffusion model was trained without access to label infomration, (b) we only use the validation set for hyperparameter optimization, and (c) the vast search space ensures that most generated compounds are neither in the training nor the validation set (see Appendix B.6 for a detailed analysis).

Guidance Model Training. For training the guidance models $f_t(\cdot;\theta)$, we use the graph convolutional neural network (GNN) architecture from Lee et al. (2023), consisting of 3 GCN layers with 16 hidden units and tanh activations (Kipf & Welling, 2016). The embeddings from each layer are concatenated and fed into two fully connected heads with tanh and sigmoid activations, respectively, whose outputs of dimension 16 are then multiplied and fed into another two-layer MLP with the same dimension and ReLU activation functions (Nair & Hinton, 2010). We train separate networks for each protein target, predicting both the most likely regression label $\boldsymbol{\mu}_t(\mathbf{g}_t;\theta) = f_t^1(\mathbf{g}_t;\theta)$, as well as log-variances $\log \boldsymbol{\sigma}_t^2(\mathbf{g}_t;\theta) = f_t^2(\mathbf{g}_t;\theta)$ that serve as an estimator of their predictive uncertainty. Given the regression labels $\mathbf{y}$, the models are optimized with respect to a negative log-likelihood loss $\mathcal{L}_t = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2\mathbf{I})$ Following the protocol of Lee et al. (2023), all models are trained for 10 epochs of stochastic gradient descent, using the Adam optimizer with a batch size of 1024 and a learning rate of 1×10^{-3} (Kingma & Ba, 2014). The corresponding objective $\mathcal{L}$ with an explicit L_2 regularization term is given by

$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}) \doteq \mathbb{E}_{p_\mathcal{T}}\left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\mathcal{T}, \boldsymbol{\sigma}_\mathcal{T}^2\mathbf{I})\right] + \frac{1}{2\lambda}||\theta||_2^2, \tag{B.19}$$

where the regularization strength λ is optimized as a hyperparameter. For weight decay regularization, we instead switch to the AdamW optimizer (Loshchilov & Hutter, 2017) and optimize the corresponding hyperparameter. Similarly, we train context-guided models by adding the additional regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ as introduced in Section 3:

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}}\left[\sum\nolimits_{j=1}^{2}\left(f_t^{(j)}(\hat{\mathbf{g}}_t;\theta) - m_t(\hat{\mathbf{g}}_t)\right)^\top K_t(\hat{\mathbf{g}}_t)^{-1}\left(f_t^{(j)}(\hat{\mathbf{g}}_t;\theta) - m_t(\hat{\mathbf{g}}_t)\right)\right], \tag{B.20}$$

Here, $\hat{\mathbf{g}}_t$ is a context batch sampled from an out-of-distribution context set of 500 000 unlabeled molecules that were randomly sampled from ZINC and processed identically to the training data. As described in Section 3, we use a randomly initialized set of parameters ϕ to generate 16-dimensional context set embeddings $h_t(\hat{\mathbf{g}}_t, \phi) \in \mathbb{R}^{M\times 16}$ and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{g}}_t) = \sigma_t h_t(\hat{\mathbf{g}}_t, \phi) h_t(\hat{\mathbf{g}}_t, \phi)^T + \tau_t \mathbf{I}.$$

Since the model returns mean and log-variance estimates $f_t^1(\hat{\mathbf{g}}_t;\theta)$ and $f_t^2(\hat{\mathbf{g}}_t;\theta)$, the target function values $m_t(\hat{\mathbf{g}}_t)$ are given by the training set mean $m_t^1(\hat{\mathbf{g}}_t) = \mathrm{mean}(\mathbf{y})$ and a log-variance hyperparameter $m_t^2(\hat{\mathbf{g}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 0.7$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx 2$ in out-of-distribution regions of the input domain. The resulting training objective is

$$\mathcal{L}^*(\theta, \mathcal{D}_\mathcal{T}) = \mathbb{E}_{p_\mathcal{T}}\left[\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T}) + R(\theta, f_\mathcal{T}, \mathcal{T}, p_{\hat{\mathbf{X}}_\mathcal{T}})\right]. \tag{B.21}$$

The covariance scale σ_t and diagonal offset τ_t are optimized as hyperparameters. To avoid overconfident predictions as context points approach the invariant distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we increase τ_t with the same schedule as the noise scales β_t, starting from $\beta_0 = \tau_t$ and ending at $\beta_T = 10\tau_t$. Similarly, as the distinction between in- and out-of-distribution becomes meaningless at larger noise levels, we decrease σ_t with an inverted schedule starting from $\beta_0 = \sigma_t$ and ending at $\beta_T = 0.1\sigma_t$.

The deep ensembles (Lakshminarayanan et al., 2017) comprise 5 independent models regularized with weight decay, which is optimized as a hyperparameter. The pre-trained model was optimized with a self-supervised denoising objective, which has been shown to be an effective pre-training method Zaidi et al. (2023). It was trained on all 500,000 unlabeled molecules from the context set for 50 epochs with the AdamW optimizer (Loshchilov & Hutter, 2017), using a batch size of 1024, a learning rate of 10^{-4}, and a weight decay of 10^{-4}. After pre-training, this model was fine-tuned on with weight decay.

All regularization hyperparameters were optimized with respect to the supervised loss $\mathcal{L}$ on the held-out, high-label test set by performing a grid search over the hyperparameter space presented in Table 3. The best hyperparameter combination was then selected and used to re-train five independent models using different random seeds.

Table 3: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.2.

Description	Hyperparameter	Search Space
L_2 regularization	λ	$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}, 10^{4}$
Single model with weight decay	λ	$10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5$
Fine-tuning with weight decay	λ	$10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5$
Ensemble with weight decay	λ	$10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5$
Context Guidance	covariance scale σ_t	$10^{-5}, 10^{-3}, 10^{-1}, 1, 10^{1}, 10^{3}$
	diagonal offset τ_t	$10^{-5}, 10^{-3}, 10^{-1}, 1, 10^{1}, 10^{3}$
	number of context points	256, 512

After retraining five independent guidance models with the best regularization hyperparameters for each of the five different proteins, we use the gradients from these models to condition the denoising process. Following Lee et al. (2023), we set the guidance strength hyperparameters to $r_{\mathbf{X},0} = 0.5$ and $r_{\mathbf{G},0} = 0$, respectively, and sample 3000 molecules each. The SA (Ertl, 2003) and QED (Bickerton et al., 2012) scores for these new molecules are derived with RDKIT (Landrum et al., 2013), while the docking score is derived with a QUICKVINA script (Alhossary et al., 2015) prepared by Lee et al. (2023).

Additional Experimental Results. We repeat the heasline results from Section 5.1 in Figure 10; following (Lee et al., 2023), we present the top 5% of normalized docking scores of compounds that are drug-like ($\mathbf{y}_{\text{SA}} < 5$ and $\mathbf{y}_{\text{QED}} > 0.5$) and novel (Tanimoto similarity of < 0.4 to the training set (Bajusz et al., 2015)). We also show the percentage of different compounds that would be classified as hits with the thresholds defined in Lee et al. (2023) in Figure 11. In addition to the main metrics from Lee et al. (2023), we also present the full distributions of:

- The objective $\mathbf{y} = \frac{\text{clip}(-\mathbf{y}_{\text{vina}}, 20, 0)}{20} \cdot \frac{10-\mathbf{y}_{\text{SA}}}{9} \cdot \mathbf{y}_{\text{QED}}$ in Figure 12.
- The clipped and normalized negative QUICKVINA score $\frac{\text{clip}(-\mathbf{y}_{\text{vina}}, 20, 0)}{20}$ in Figure 13.
- The normalized synthetic accessibility (SA) score $\frac{10-\mathbf{y}_{\text{SA}}}{9}$ in Figure 14.
- The quantitative estimate of drug-likeness (QED) score $\mathbf{y}_{\text{QED}}$ in Figure 15.

Furthermore, we normalize the scores of the generated compounds by their molecular weight and present the corresponding results in Figure 16, ensuring the performance differences are not caused by the bias of the docking score toward larger molecules. Finally, we show the same models trained on noisy labels $\bar{\mathbf{y}} = \mathbf{y} \cdot \epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ in Figure 17. Finally, the full distribution of models trained on an 85%-15% i.i.d. split is presented in Figure 18.

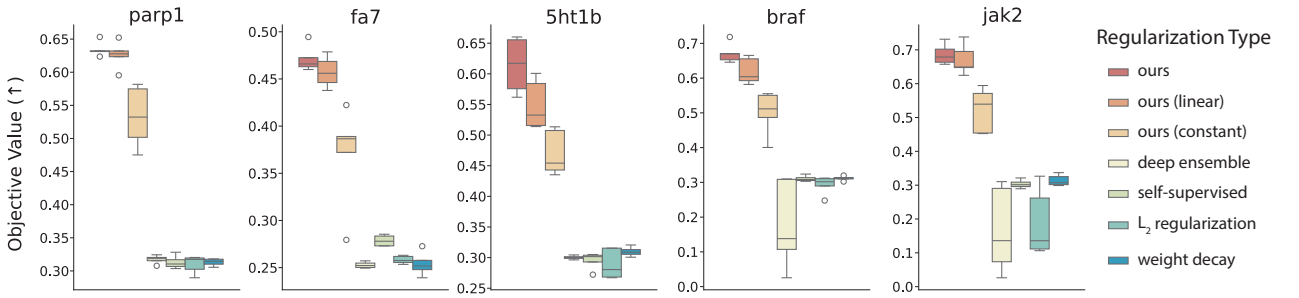


Figure 10: Comparison of the samples generated with different guided diffusion models across five distinct protein targets using the normalized top-5% docking score of novel and drug-like compounds as in Lee et al. (2023).

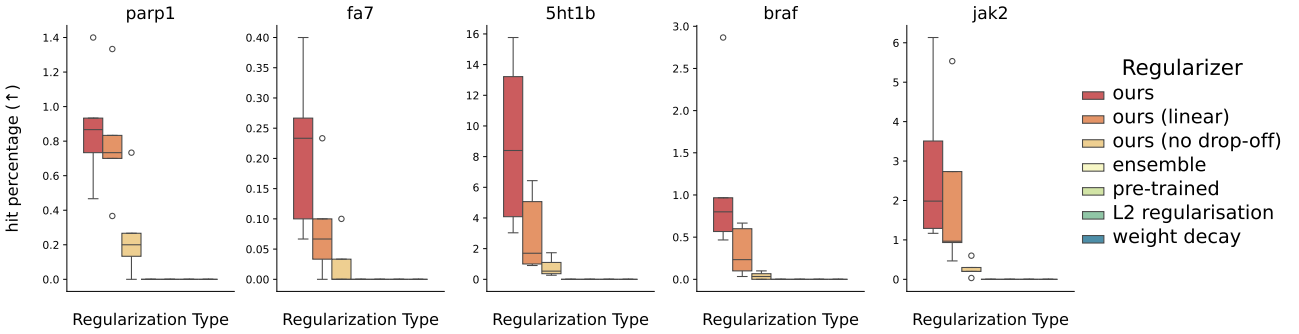


Figure 11: The percentage of small molecules denoted as hits using the thresholds from Lee et al. (2023), given by an unnormalized negative docking score of 10 for PARP1, 8.5 for FA7, 8.7845 for 5HT1B, 9.1 for JAK2, and 10.3 for BRAF.

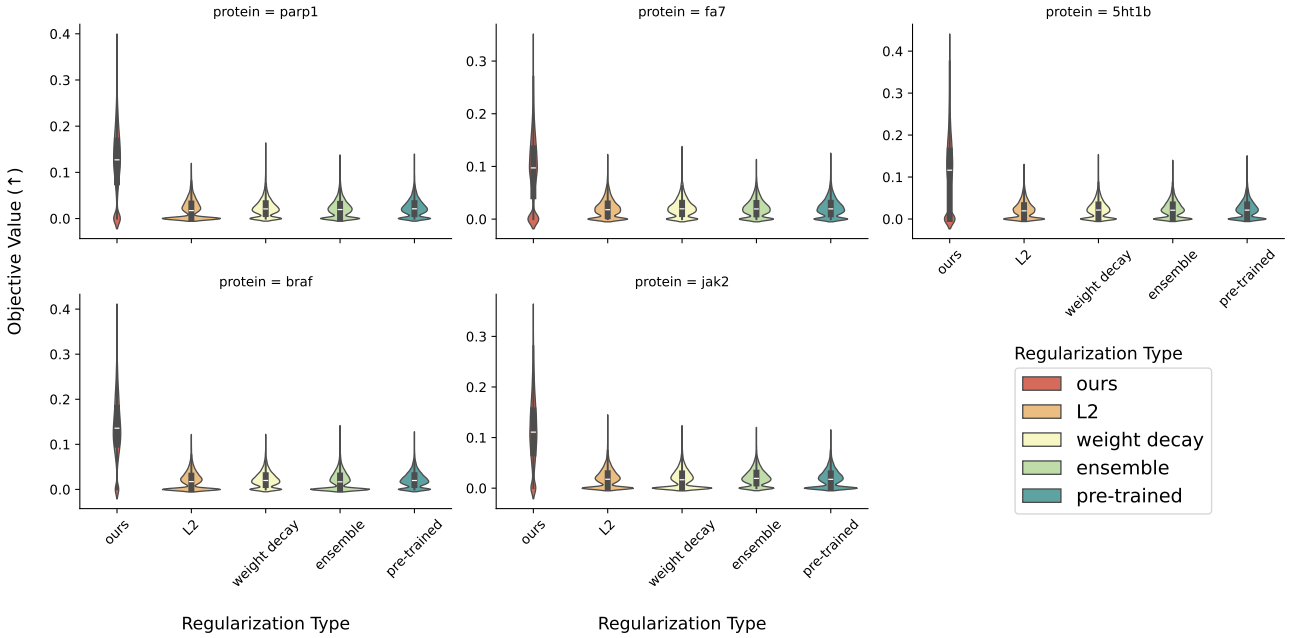


Figure 12: Full distributions of the objective values $\mathbf{y}$ of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

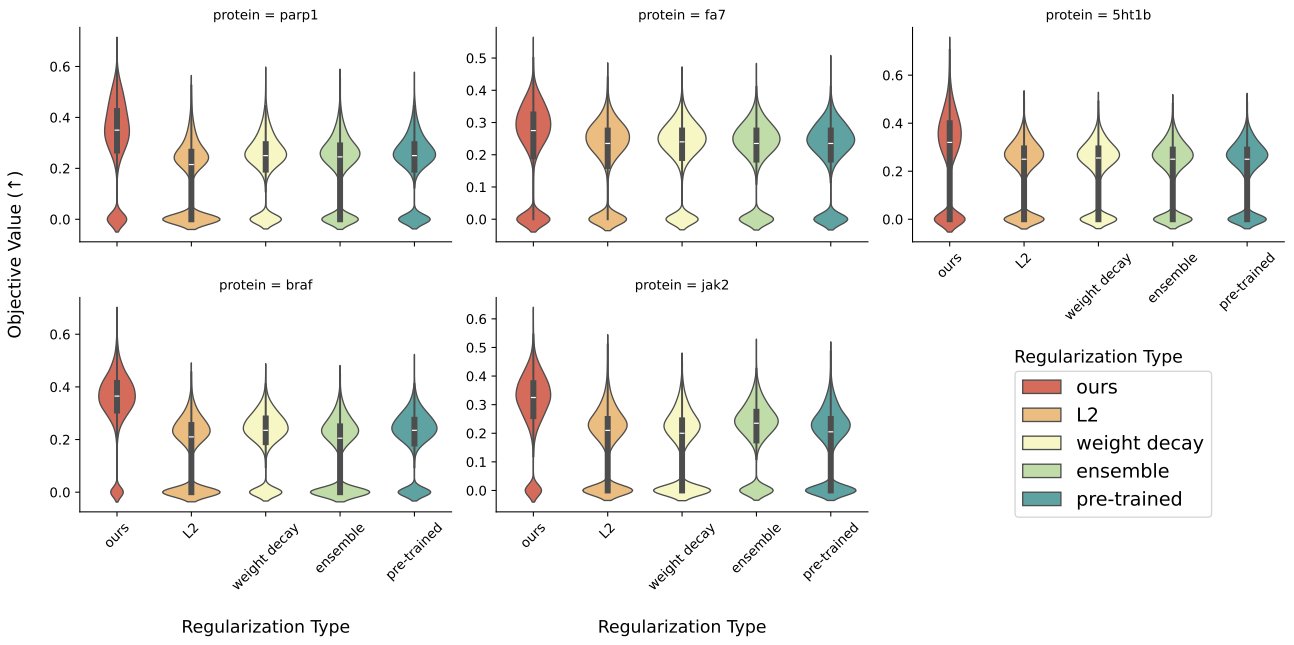


Figure 13: Full distributions of the clipped and normalized negative QUICKVINA docking score $\mathbf{y}_{\text{vina}}$ (Alhossary et al., 2015) of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

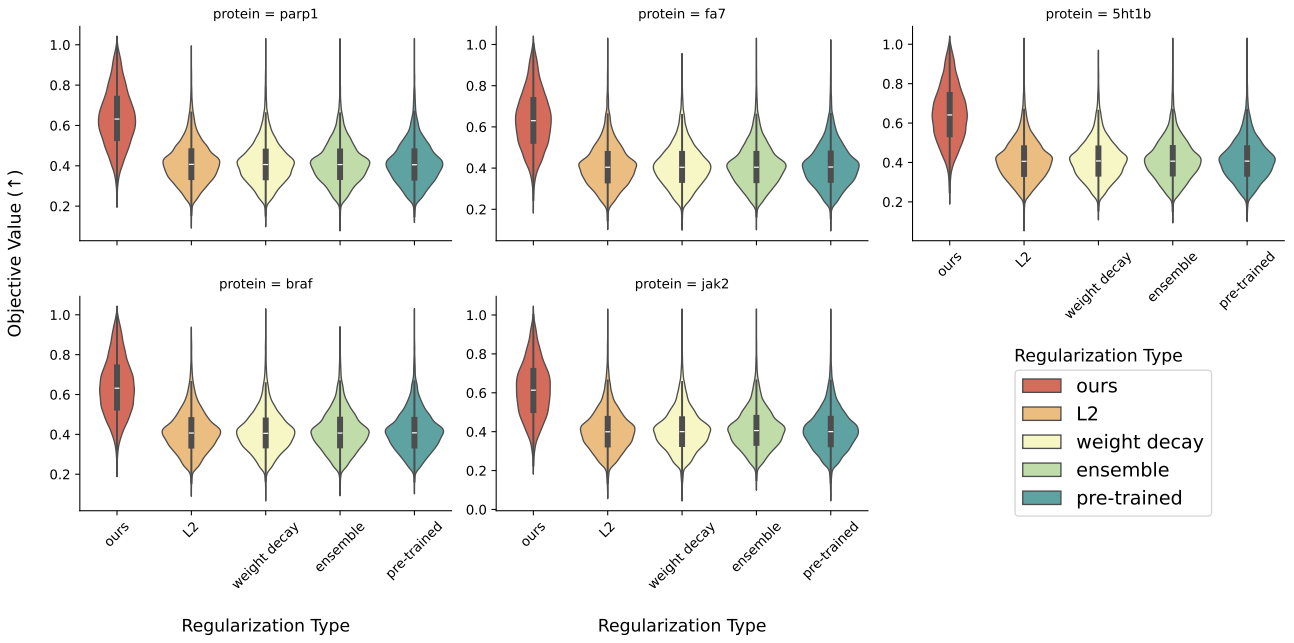


Figure 14: Full distributions of the normalized synthetic accessibility (SA) score $\mathbf{y}_{\text{SA}}$ (Ertl, 2003) of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

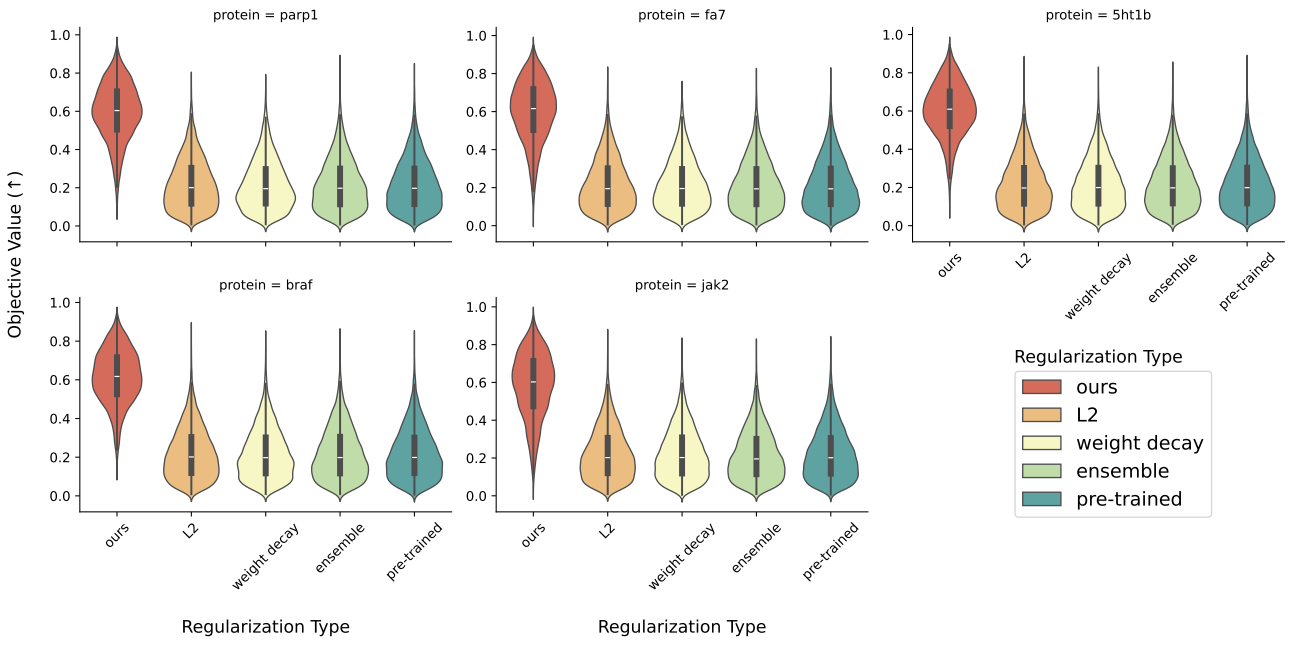


Figure 15: Full distributions of the quantitative estimate for drug-likeness (QED) score $\mathbf{y}_{\text{QED}}$ of all generated compounds for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

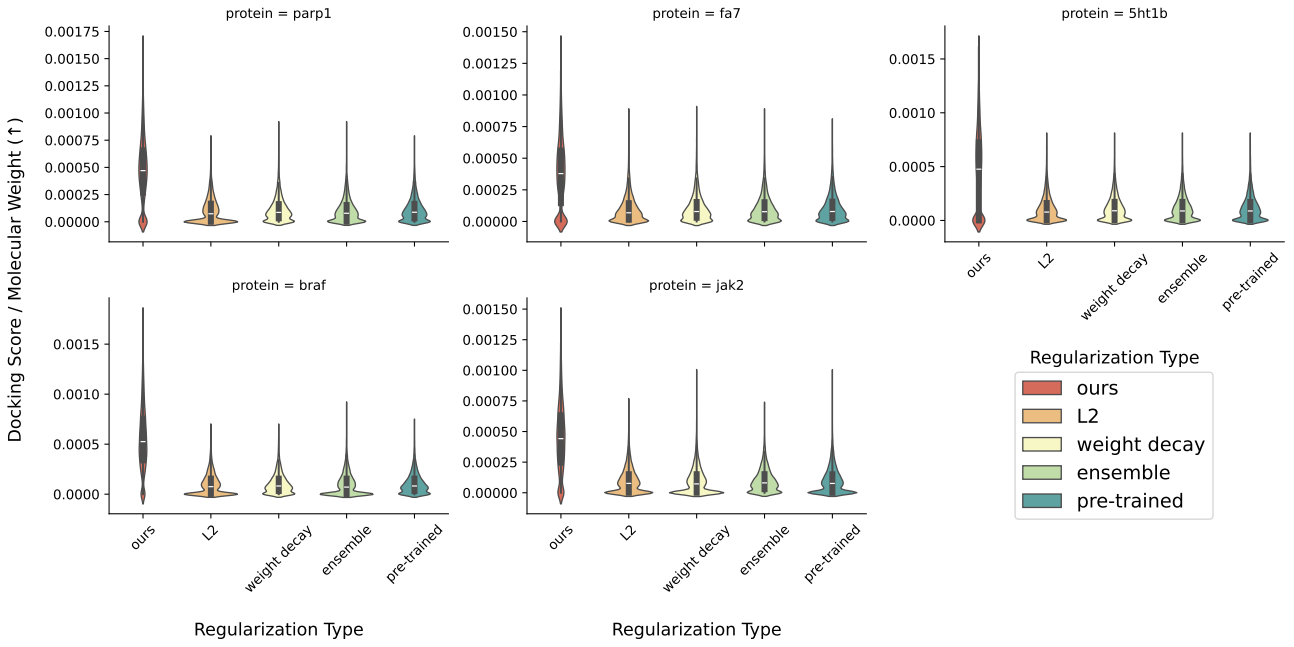


Figure 16: Full distributions of the objective values $\mathbf{y}$ of all generated compounds normalized by their molecular weight for each protein target across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

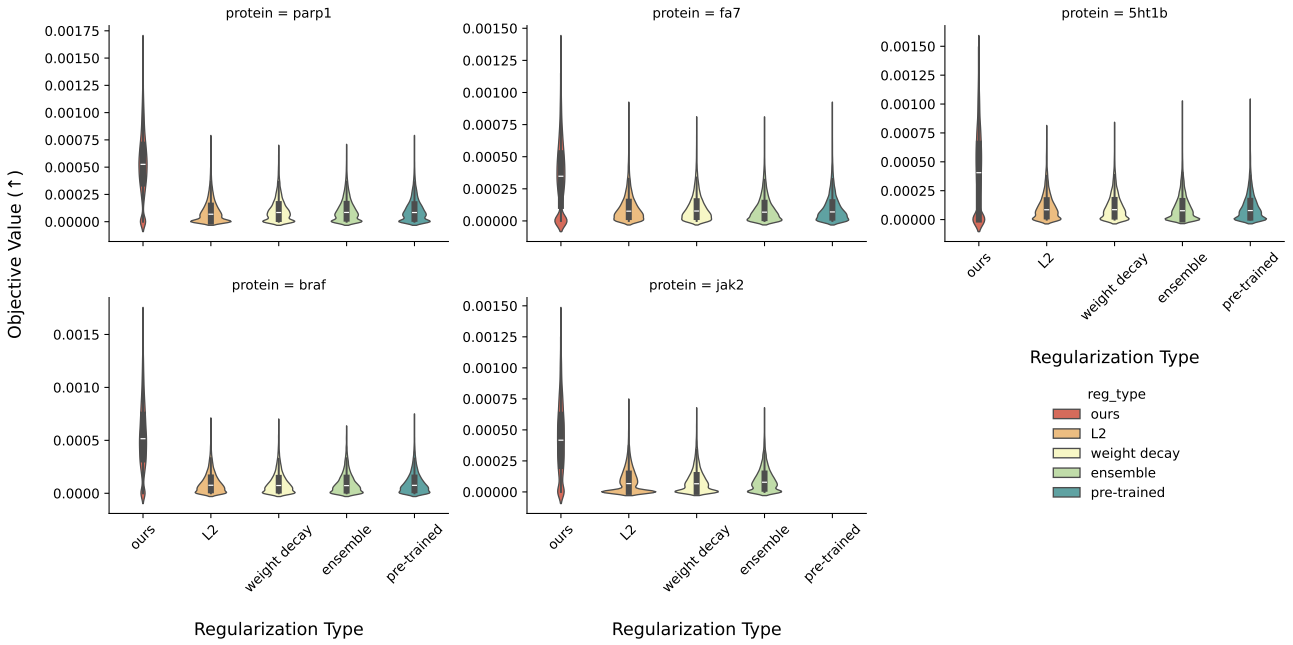


Figure 17: Full distributions of the objective values $\mathbf{y}$ of all generated compounds when models are trained on noised labels $\bar{y} = \mathbf{y} \cdot \epsilon$ with added noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, across five independent training and sampling runs with different random seeds. These results mirror those in Figures 4 and 10.

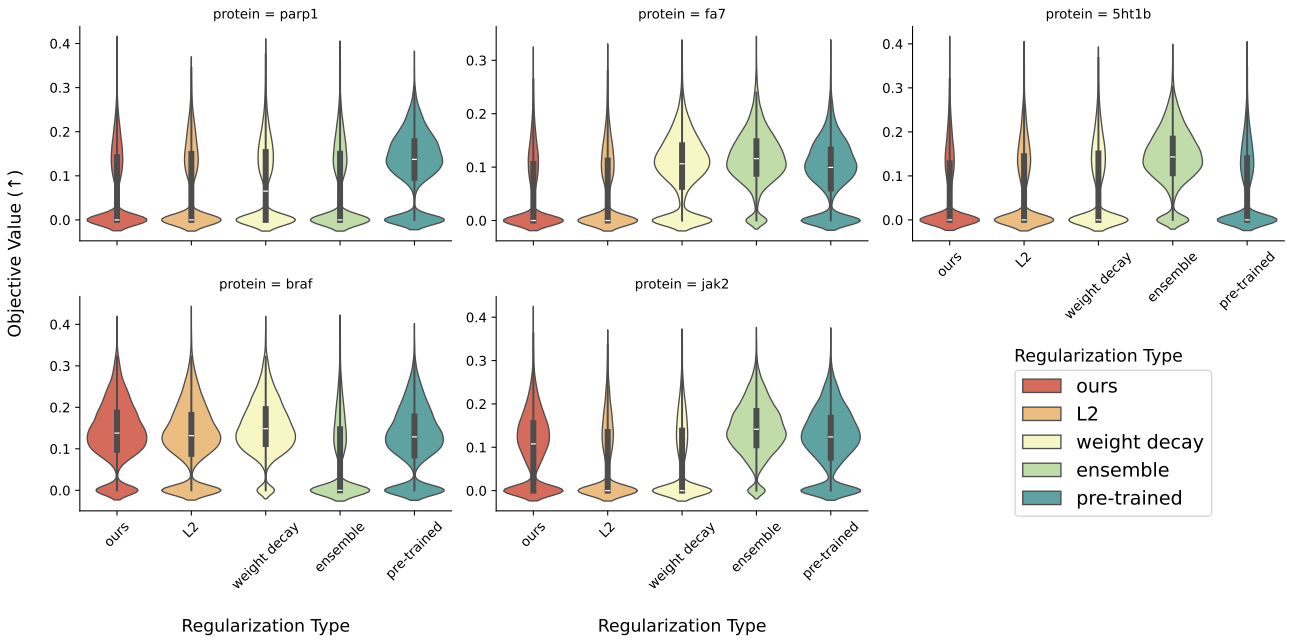


Figure 18: Full distributions of the objective values $\mathbf{y}$ of all generated compounds when guidance models that are trained on an 85%-15% i.i.d. random split, across five independent training and sampling runs with different random seeds. As expected, the deep ensembles and pre-trained methods perform particularly well in this setting. Interestingly, the performance of our domain-aware guidance method stays approximately the same between the random and label-split evaluation settings.

Comparison to Unsupervised Domain Adaptation Techniques. In addition to the models presented above, we also compare our method to guidance functions that are regularized with three robust and well-established unsupervised domain adaptation techniques, namely:

- DEEPCORAL: a method that aligns the feature distributions of the last-layer guidance model embeddings across the labeled data and context set through an auxiliary loss term (Sun & Saenko, 2016).
- Domain-Adversarial Neural Networks (DANN): a gradient-reversal approach that aligns the labeled data and context set embeddings by subtracting the gradients of a linear domain classifier from those of the guidance model (Ganin et al., 2016).
- DOMAIN CONFUSION: a method that tries to learn domain-invariant representations by maximizing the predictive entropy of an adversarially trained domain classifier (Tzeng et al., 2015).

We adapted the DEEPCORAL and DANN implementations from the DrugOOD library (Ji et al., 2023) and implemented the DOMAIN CONFUSION models from scratch, closely following the description in Tzeng et al. (2015). All models were evaluated with the same training, hyperparameter selection, and sample generation protocols as in Appendix B.2, using the domain adaptation hyperparameter search spaces presented in Table 4.

Table 4: Hyperparameter search space for the unsupervised domain adaptation algorithms.

Method	**Hyperparameter**	**Search Space**
DEEPCORAL	loss weight λ	10^{-2}, 5×10^{-2}, 10^{-1}, 5×10^{-1}, 1, 5, 10^{1}, 5×10^{1}, 10^{2}
DANN	reversal factor α	10^{-2}, 5×10^{-3}, 10^{-3}, 5×10^{-4}, 10^{-4}, 5×10^{-5}
DOMAIN CONFUSION	loss weight λ	10^{-2}, 5×10^{-2}, 10^{-1}, 5×10^{-1}, 1, 5, 10^{1}, 5×10^{1}, 10^{2}

The resulting performance metrics for the first three protein targets from Lee et al. (2023) (PARP1, FA7, and 5HT1B) are presented in Table 5. While these methods consistently outperform the guidance models trained with standard approaches such as L_2 regularization and weight decay, they still perform considerably worse than our approach. One possible explanation for this performance gap is that the DEEPCORAL, DANN, and DOMAIN CONFUSION methods only act on the embedding space of a guidance model without considering the behavior of its predictions. In contrast, our context-guided approach directly regularizes the model's predictions, encouraging it to learn functions that revert to an uninformative prior in regions that are far from the training data. This allows our method to better capture the underlying structure of the data and improves its robustness in out-of-distribution settings.

Table 5: Performance comparison of our method against guidance models regularized with unsupervised domain adaptation techniques (DEEPCORAL, DANN, and DOMAIN CONFUSION) and standard regularization approaches (WEIGHT DECAY) on the first three protein targets from Lee et al. (2023) (PARP1, FA7, and 5HT1B). Our context-guided regularization scheme outperforms all domain adaptation techniques while maintaining comparable computational efficiency. Results are reported as the mean and standard deviation across five independent training and sampling runs with different random seeds. The best results for each target are shown in bold.

Regularizer	Average batch time (s)	PARP1 (↑)	FA7 (↑)	5HT1B (↑)
Ours	$0.08_{\pm 0.00}$	$\mathbf{0.64_{\pm 0.01}}$	$\mathbf{0.47_{\pm 0.00}}$	$\mathbf{0.60_{\pm 0.03}}$
DEEPCORAL	$0.08_{\pm 0.00}$	$0.36_{\pm 0.01}$	$0.32_{\pm 0.00}$	$0.35_{\pm 0.04}$
DANN	$0.07_{\pm 0.00}$	$0.37_{\pm 0.07}$	$0.27_{\pm 0.03}$	$0.34_{\pm 0.07}$
DOMAIN CONFUSION	$0.08_{\pm 0.00}$	$0.40_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.38_{\pm 0.01}$
WEIGHT DECAY	$0.02_{\pm 0.00}$	$0.31_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.31_{\pm 0.01}$

Context Set Sensitivity Studies. As the context set provides our method with structured information about a broader subset of the input domain, we found that its size and composition are key factors in determining model performance. To investigate this relationship, we carried out a series of sensitivity studies for the fist three protein targets from Lee et al. (2023) (PARP1, FA7, and 5HT1B). All experiments were performed with the same experimental setup, hyperparameter search space, and evaluation protocol as described in Appendix B.2, differing only in the choice of context set. The full results are presented in Table 6

First, we compare the performance of a model trained with the full context set (100%) to models trained with randomly sampled subsets of size 10% and 1%, respectively. Mirroring the results from Section 5.2, we observe a strong deterioration in performance across these settings, with the performance of the models trained with the smallest context set (1%) reverting to that of models trained with standard weight decay.

To explore whether an even larger and more diverse context set could lead to further performance gains, we trained an additional set of models using the 665k biologically and pharmaceutically relevant molecules from the QMUGS dataset (Isert et al., 2022), originally curated from the CHEMBL database (Gaulton et al., 2017). While we observe some performance improvements, they are relatively marginal, indicating that our original context set was likely already at or past the point of diminishing returns regarding its size.

Finally, we performed an additional sensitivity study for which we selected the 10% of molecules in the original context set that were either most or least similar to the labeled training data (as measured by their maximum ECFP4-based Tanimoto similarity, calculated with rdkit (Landrum et al., 2013)). We note that the average maximum Tanimoto similarity between these context sets and the training data is still relatively low, at 0.42 ± 0.06 and 0.31 ± 0.04, respectively. We observe that using a much smaller context set with near-OOD molecules (most similar 10%) is able to recover the performance of using the full context set (100%), while relying on less relevant molecules (least similar 10%) leads to worse performance than a random subsample (10%).

Table 6: Impact of the size and composition of the context set on the performance of our proposed regularizer for the small molecules application presented in Section 5.1. We compare the full context set (100%) to randomly sampled subsets (10% and 1%), as well as subsets of the original context set containing the 10% of molecules that are either most or least similar to the labeled training data. We additionally train models with an even larger and more diverse context set derived from QMUGS (Isert et al., 2022). Results are reported as the mean and standard deviation across five independent training and sampling runs with different random seeds.

Regularizer	parp1 (↑)	fa7 (↑)	5ht1b (↑)
ours (100%)	$0.64_{\pm 0.01}$	$0.47_{\pm 0.00}$	$0.60_{\pm 0.03}$
ours (10%)	$0.54_{\pm 0.01}$	$0.45_{\pm 0.00}$	$0.44_{\pm 0.04}$
ours (1%)	$0.37_{\pm 0.07}$	$0.27_{\pm 0.03}$	$0.30_{\pm 0.07}$
ours (QMUGS)	$0.69_{\pm 0.02}$	$0.47_{\pm 0.00}$	$0.62_{\pm 0.02}$
ours (most similar 10%)	$0.66_{\pm 0.01}$	$0.45_{\pm 0.01}$	$0.59_{\pm 0.01}$
ours (least similar 10%)	$0.51_{\pm 0.02}$	$0.32_{\pm 0.04}$	$0.42_{\pm 0.01}$
weight decay	$0.31_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.31_{\pm 0.01}$

B.3. Equivariant Diffusion for Materials.

Dataset. We build on the experimental setup from Weiss et al. (2023) and use a dataset of all 34071 cata-condensed polybenzenoid hydrocarbons (cc-PBHs) with up to 11 benzene rings (Wahab et al., 2022). In contrast to the setup in Section 5.1 and Appendix B.2, which uses a subset of the billions of compounds from the ZINC database (Tingle et al., 2023) to represent the broader chemical space of up to 1×10^{60} drug-like small molecules (Bohacek et al., 1996; Ertl, 2003), this application operates in a closed and fully enumerated input space. This allows us to construct data splits that precisely control how much generalization is required to reach held-out regions with optimal objective values.

Starting from the GFN2-xTB optimized ground-state cc-PBH structures, we adopt the graph-of-rings representation from Weiss et al. (2023), representing a compound with R rings as a point cloud of ring centroids $\mathbf{x}_t \in \mathbb{R}^{R\times 3}$. While computing electronic properties requires 3D representations, much of the combinatorial structure of this data persists in 2D topological graphs. We thus generate extended reduced graph fingerprints (Stiefl et al., 2006) in RDKIT (using a fuzziness increment of 0 and a maximum path length of 50), which correspond to a 2D equivalent of the 3D graph-of-rings framework. These fingerprints are binarized and compared via the Jaccard-Tanimoto kernel (Bajusz et al., 2015) to measure pairwise similarities. This enables us to cluster the dataset and derive maximally distinct training, validation, test, and context sets (Figure 19). Since the framework of Weiss et al. (2023) only generates compounds with a predetermined number of rings R and 75% of the dataset consists of molecules with $R = 11$, we split the data as follows: All compounds with $R < 11$ are used as part of an out-of-distribution context set. The remaining compounds are split into 3 groups using spectral clustering (scikit-learn) (Pedregosa et al., 2011). The largest cluster of 16 973 compounds is used as the training set. The second largest cluster of 4538 compounds is used as the held-out test set. The smallest cluster of 3883 compounds is used as the validation set for hyperparameter selection and early stopping. When training models with our context-guided regularization scheme, we use either all molecules with $R < 11$ as a *reduced* context set or all compounds not in the training set as a *full* context set

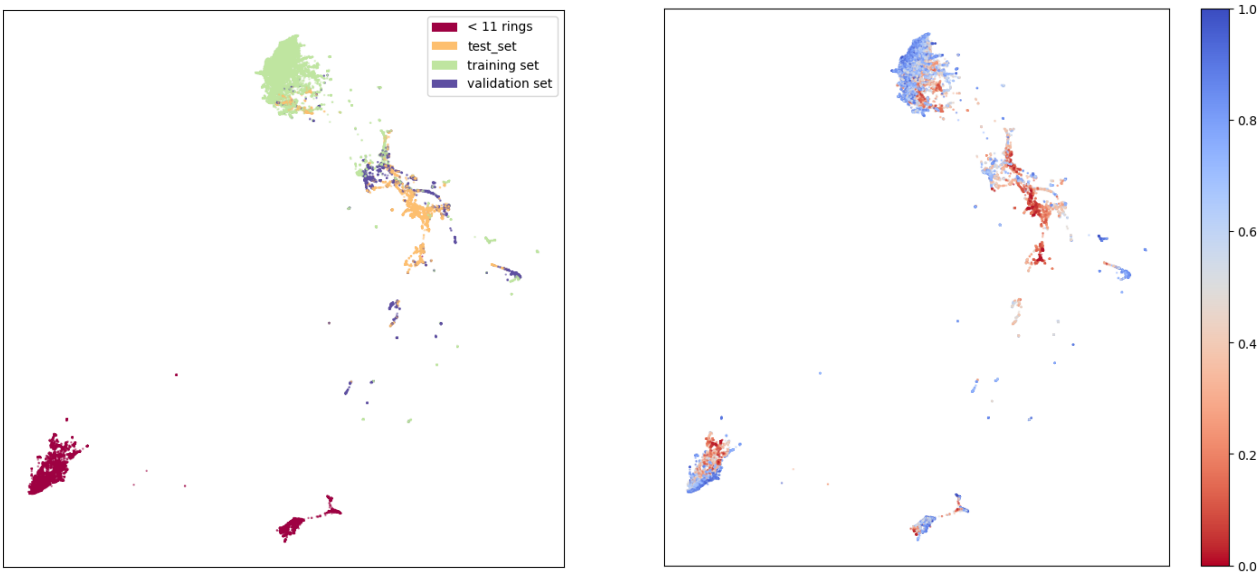


Figure 19: UMAP dimensionality reduction plots (McInnes et al., 2018) of the cc-PBH dataset using extended reduced graph fingerprints (Stiefl et al., 2006) and the Jaccard-Tanimoto distance. **Left**: Points are color-coded according to whether they belong to the out-of-distribution context set with fewer than 11 rings or the training, validation, and test set. **Right:** The same points, color-coded according to the composite objective used by Weiss et al. (2023) and defined as $\mathbf{y} = 3\,\mathbf{y}_{\text{HLG}} + \mathbf{y}_{\text{IP}} - \mathbf{y}_{\text{EA}}$. Lower values indicate more desirable properties and are, for $R = 11$, concentrated in the cluster used as a held-out test set.

Each molecule in the dataset used by Weiss et al. (2023) is annotated with five electronic properties: LUMO energy ($\mathbf{y}_{\text{LUMO}}$), HOMO-LUMO gap ($\mathbf{y}_{\text{HLG}}$), relative energy ($\mathbf{y}_{\text{REL}}$), adiabatic ionization potential ($\mathbf{y}_{\text{IP}}$), and adiabatic electron affinity ($\mathbf{y}_{\text{EA}}$) (Wahab et al., 2022). The full property distributions for each data split are shown in Figure 20. Conveniently, the held-out test set cluster contains compounds with significantly better $\mathbf{y}_{\text{HLG}}$, $\mathbf{y}_{\text{IP}}$, and $\mathbf{y}_{\text{EA}}$ values than the training set, which are the properties used to define the composite objective in Weiss et al. (2023), as is also apparent in Figure 19.

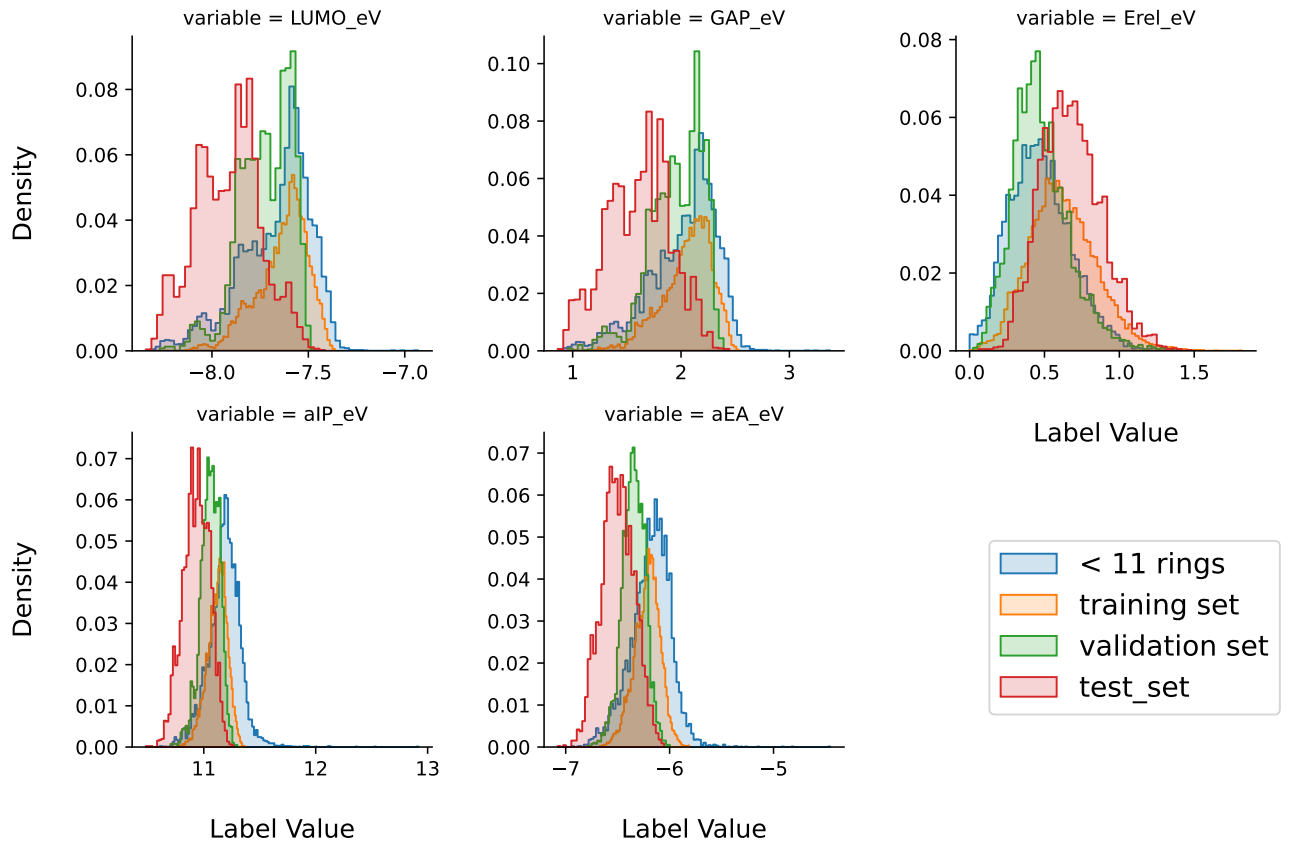


Figure 20: The full electronic property distributions for each data split.

Diffusion Model Training. Following Weiss et al. (2023), we train an E(3)-equivariant diffusion model (EDM; Hoogeboom et al. (2022)) on the graph-of-rings-derived point clouds $\mathbf{x}_t \in \mathbb{R}^{R\times 3}$. Both the forward noising and reverse denoising process outlined in Section 2 are made equivariant to permutations, translations, rotations, and reflections by using an E(3)-equivariant graph neural network (E(3)-GNN) (Satorras et al., 2021) as the score model and redefining a translation-invariant noising process in the linear subspace of $\sum_i \mathbf{x}_i = 0$ (Hoogeboom et al., 2022).

We optimized the resulting equivariant diffusion model on the 16 973 compounds in the training set using the same hyperparameter settings as in Weiss et al. (2023). Specifically, the score network $s_\theta(\mathbf{x}_t, t)$ is given by a 9-layer E(3)-GNN with 192 hidden units and $\tanh$ activations. It is trained for 1000 epochs with the ADAM optimizer (Kingma & Ba, 2014) using a learning rate of 1×10^{-3}, a batch size of 256, and a small weight decay hyperparameter of 10^{-12}. The diffusion process is given by a cosine β_t schedule (Nichol & Dhariwal, 2021) with $T = 1000$ time-steps, and $\beta_0 = 0$ and $\beta_T = 1$. Due to the reduced problem space of the graph-of-rings representation, we find that the vast majority of generated point clouds ($\geq 99\%$, as in Weiss et al. (2023)) correspond to valid molecules and can be parsed by RDKIT (Landrum et al., 2013).

Guidance Model Training. Similar to the score network, the guidance model uses a 12-layer E(3)-GNN architecture with 192 hidden units and $\tanh$ activations, matching the model hyperparameters from Weiss et al. (2023). These models are trained using the ADAM optimizer (Kingma & Ba, 2014) for up to 1000 epochs with a learning rate of 6×10^{-4} and batch size of 256, stopping if the validation set performance deteriorates for more than 10 consecutive epochs.

As in Weiss et al. (2023), we train these models as multi-property prediction networks, outputting both the regression mean $\boldsymbol{\mu}_t(\hat{\mathbf{x}}_t; \theta) = f_t^1(\hat{\mathbf{x}}_t; \theta)$ and log-variance $\log \boldsymbol{\sigma}_t^2(\hat{\mathbf{x}}_t; \theta) = f_t^2(\hat{\mathbf{x}}_t; \theta)$ for all five properties $\mathbf{y}_{\text{HLG}}$, $\mathbf{y}_{\text{IP}}$, $\mathbf{y}_{\text{LUMO}}$, $\mathbf{y}_{\text{REL}}$, and $\mathbf{y}_{\text{EA}}$. Given the true regression labels $\mathbf{y}$, the models are trained by minimizing to a negative log-likelihood loss across these five properties $\mathcal{L}_t = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \mathbf{I})$.

The corresponding objective $\mathcal{L}$ with an explicit L_2-regularization term is given by

$$\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}) \doteq \mathbb{E}_{p_\mathcal{T}} \left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\mathcal{T}, \boldsymbol{\sigma}_\mathcal{T}^2 \mathbf{I}) \right] + \frac{1}{2\lambda} ||\theta||_2^2, \tag{B.22}$$

where the regularization strength λ is optimized as a hyperparameter. Similarly, models are regularized with weight decay by switching to the ADAMW optimizer (Loshchilov & Hutter, 2017) and optimizing the corresponding hyperparameter. Context-guided models are trained with the regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ as introduced in Section 3

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}} \left[\sum_{j=1}^{2} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right)^\top K_t(\hat{\mathbf{x}}_t)^{-1} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right) \right], \tag{B.23}$$

where $\hat{\mathbf{x}}_t \sim p_{\hat{\mathbf{X}}_t}(\hat{\mathbf{x}}_t)$ is a context batch sampled from either the *full* context set covering all compounds not in the training set, or the *reduced* context set containing only compounds with fewer than 11 rings (see Figure 19). As described in Section 3, we use a randomly initialized parameter set ϕ to generate 196-dimensional context set embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M \times 196}$ from the E(3)-GNN's last layer and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{x}}_t) = \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}.$$

Since the model predicts means and log-variances $f_t^1(\hat{\mathbf{x}}_t; \theta)$ and $f_t^2(\hat{\mathbf{x}}_t; \theta)$, the target function values $m_t(\hat{\mathbf{x}}_t)$ are given by the training set mean and a log-variance hyperparameter $m_t^2(\hat{\mathbf{x}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 1$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx e$ in out-of-distribution regions of the input domain. The resulting training objective is

$$\mathcal{L}^*(\theta, \mathcal{D}) = \mathbb{E}_{p_\mathcal{T}} \left[\mathcal{L}(\theta, \mathcal{D}_\mathcal{T}, \mathcal{T}) + R(\theta, f_\mathcal{T}, \mathcal{T}, p_{\hat{\mathbf{X}}_\mathcal{T}}) \right]. \tag{B.24}$$

The covariance scale σ_t and diagonal offset τ_t are optimized as hyperparameters. To avoid overconfident predictions as context points approach the invariant distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we increase τ_t with the same schedule as the noise scales β_t, starting from $\beta_0 = \tau_t$ and ending at $\beta_T = 10\tau_t$. Similarly, as the distinction between in- and out-of-distribution becomes meaningless at larger noise levels, we decrease σ_t with an inverted schedule starting from $\beta_0 = \sigma_t$ and ending at $\beta_T = 0.1\sigma_t$. The regularization hyperparameters are optimized via grid search on the held-out validation set, using the supervised loss $\mathcal{L}$ as the performance metric. The grid search was over the hyperparameter space shown in Table 7. The best hyperparameter combination was selected and used to re-train ten independent models with different random seeds.

Table 7: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.3.

Description	Hyperparameter	Search Space
L_2 regularization	λ	10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^{-3}
Weight decay	λ	10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.5
Context Guidance	covariance scale σ_t	10^{-2}, 10^{-1}, 1, 10^1, 10^{-2}
	diagonal offset τ_t	10^{-2}, 10^{-1}, 1, 10^1, 10^{-2}
	number of context points	16, 64, 256

After retraining ten independent guidance models for each regularization type, their predictions for the HOMO-LUMO gap ($\mathbf{y}_{\text{HLG}}$), adiabatic ionization potential ($\mathbf{y}_{\text{IP}}$), and adiabatic electron affinity ($\mathbf{y}_{\text{EA}}$) properties were used to define the composite objective $\mathbf{y} = 3\,\mathbf{y}_{\text{HLG}} + \mathbf{y}_{\text{IP}} - \mathbf{y}_{\text{EA}}$. The resulting gradients were then used to guide the generation of 512 samples each, using an additional guidance scale s

$$\nabla_\mathbf{X} \log p_t(\mathbf{x}_t \mid \mathbf{y}) = \nabla_\mathbf{X} \log p_t(\mathbf{x}_t) + \nabla_\mathbf{X} \log p_t(\mathbf{y} \mid \mathbf{x}_t)^s = s_\theta(\mathbf{x}_t, t) + s \cdot \nabla_\mathbf{X} \log f_t(\mathbf{x}_t; \theta)$$

Additional Experimental Results. In addition to the objective values in Figure 5, we also calculate the proportion of generated samples that are (1) valid and novel (i.e. not in the training set) in Figure 21 and (2) valid and unique (i.e. only generated once) in Figure 22. Together with Figure 5, the results suggest that as guidance strength increases, our method generates more novel molecules with better objective values, yet less diversity. This may be explained by stronger guidance signals causing the denoising process to converge to the same high-value subsets of the input domain.

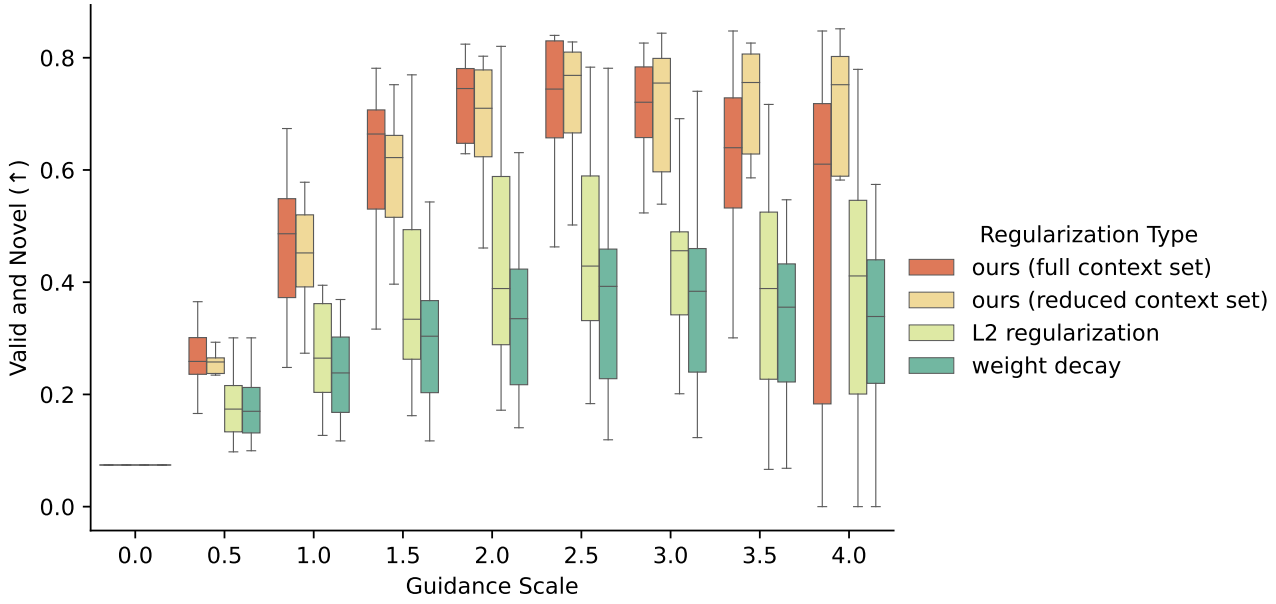


Figure 21: Performance comparison of guidance models with different regularization schemes across ten independent training and sampling runs. The proportion of compounds that are both valid and novel (i.e. not in the training set) is computed for each run and aggregated across random seeds.

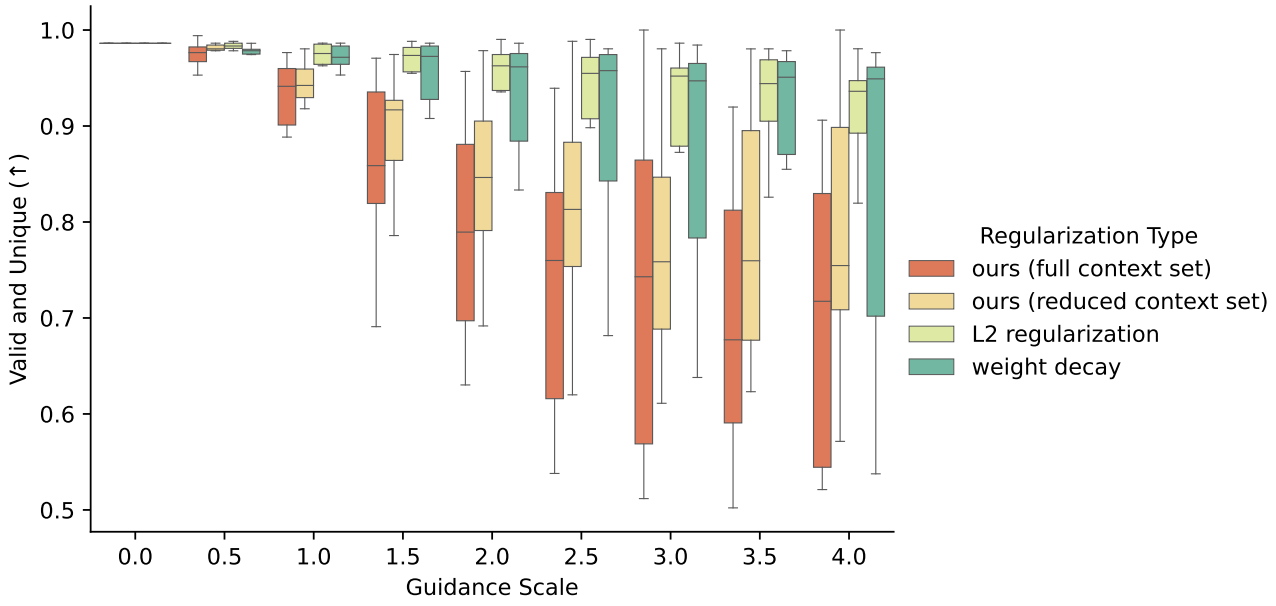


Figure 22: Performance comparison of guidance models with different regularization schemes across ten independent training and sampling runs. The proportion of compounds that are both valid and unique (i.e. only generated once) is computed for each run and aggregated across random seeds.

B.4. Discrete Diffusion for Protein Sequences.

Dataset. We use the same antibody sequence dataset as Gruver et al. (2023), consisting of approximately 100 000 paired heavy and light chain sequences from the OAS and SAbDab databases (Olsen et al., 2022; Dunbar et al., 2014) aligned with ANARCI (Dunbar & Deane, 2016) and padded to length 300. We focus on the synthetic-label applications presented by Gruver et al. (2023), namely the single-objective optimization of a protein's solvent-accessible surface area (SASA) and percentage of beta sheets. In the following experiments we optimize the latter, using code from Gruver et al. (2023) and the BioPython library (Cock et al., 2009) to derive the labels $\mathbf{y}$. Similar to Appendices B.2 and B.3, we split the data into a low-property training set and a high-property validation set to optimize for hyperparameters that perform well in label-shifted, out-of-distribution settings. Additionally, a random subsample of a third of the dataset is split off as a held-out unlabeled context set for our context-guided method, ordering the remaining two-thirds by their labels $\mathbf{y}$ and splitting them in half to generate the low-value training and high-value validation sets.

Diffusion and Guidance Model Training. In contrast to Sections 5.1 and 5.2, the guidance model of Gruver et al. (2023) is not a standalone model, but rather a regression head of the diffusion model's score network, trained jointly with the denoising objective. Specifically, a masked language model (Austin et al., 2021) learns sequence embeddings, from which two MLPs estimate the score and properties. Following Gruver et al. (2023), we use a BERT-based transformer backbone (Bhargava et al., 2021; Devlin et al., 2018) with 4 layers and 512 hidden units. The score head is a single linear layer, while the property prediction head is a feedforward network with one 512-unit hidden layer and $\tanh$ activations. This model is trained for 500 epochs with AdamW (Loshchilov & Hutter, 2017), using a batch size of 128 and a learning rate of 5×10^{-4} with a 10-step linear warmup schedule. The noising process is given by a discrete formulation from Gruver et al. (2023), using cosine β_t schedule (Nichol & Dhariwal, 2021) with $T = 1000$ time-steps, and $\beta_0 = 0$ and $\beta_T = 1$.

We train the regression head to estimate both the regression mean $\boldsymbol{\mu}_t(\hat{\mathbf{x}}_t; \theta) = f_t^1(\hat{\mathbf{x}}_t; \theta)$ and log-variance $\log \boldsymbol{\sigma}_t^2(\hat{\mathbf{x}}_t; \theta) = f_t^2(\hat{\mathbf{x}}_t; \theta)$. Given the true regression labels $\mathbf{y}$, the property prediction model is optimized with respect to the corresponding negative log-likelihood loss:

$$\mathcal{L} = -\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\sigma}_{\mathcal{T}}^2 \mathbf{I})$$

Following (Gruver et al., 2023), we take one gradient step with respect to $\mathcal{L}$ for every 5 gradient steps with respect to the score matching objective, logging the validation loss every 5 epochs and loading the best checkpoint for sample generation. We train L_2-regularized models by augmenting the supervised objective $\mathcal{L}$ with an explicit L_2 penalty

$$\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}) \doteq \mathbb{E}_{p_{\mathcal{T}}} \left[-\log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathcal{T}}, \boldsymbol{\sigma}_{\mathcal{T}}^2 \mathbf{I}) \right] + \frac{1}{2\lambda} ||\theta||_2^2, \tag{B.25}$$

where the regularization strength λ is optimized as a hyperparameter. Similarly, models are regularized with weight decay by switching to the AdamW optimizer (Loshchilov & Hutter, 2017) and optimizing the corresponding hyperparameter.

Context-guided models are trained by adding the additional regularization term $R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t})$ given by

$$R(\theta, f_t, t, p_{\hat{\mathbf{X}}_t}) = \mathbb{E}_{p_{\hat{\mathbf{X}}_t}} \left[\sum_{j=1}^{2} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right)^\top K_t(\hat{\mathbf{x}}_t)^{-1} \left(f_t^{(j)}(\hat{\mathbf{x}}_t; \theta) - m_t(\hat{\mathbf{x}}_t) \right) \right], \tag{B.26}$$

where $\hat{\mathbf{x}}_t \sim p_{\hat{\mathbf{X}}_t}(\hat{\mathbf{x}}_t)$ is a context batch sampled from the unlabeled context set split off as detailed before. As described in Section 3, we use a randomly initialized parameter set ϕ to generate 512-dimensional context set embeddings $h_t(\hat{\mathbf{x}}_t, \phi) \in \mathbb{R}^{M \times 512}$ from the last layer of the regression head and use them to construct the covariance matrix

$$K_t(\hat{\mathbf{x}}_t) = \sigma_t h_t(\hat{\mathbf{x}}_t, \phi) h_t(\hat{\mathbf{x}}_t, \phi)^T + \tau_t \mathbf{I}.$$

Since the model predicts mean and log-variance estimates $f_t^1(\hat{\mathbf{x}}_t; \theta)$ and $f_t^2(\hat{\mathbf{x}}_t; \theta)$, the target function values $m_t(\hat{\mathbf{x}}_t)$ are given by the training set mean and a log-variance hyperparameter $m_t^2(\hat{\mathbf{x}}_t) = \boldsymbol{\sigma}_0^2$, respectively. The latter is set to $\boldsymbol{\sigma}_0 = 1$, chosen to induce high predictive uncertainty estimates of $\exp(\boldsymbol{\sigma}^2) \approx e$ in out-of-distribution regions of the input domain. The resulting training objective is

$$\mathcal{L}^*(\theta, \mathcal{D}) = \mathbb{E}_{p_{\mathcal{T}}} \left[\mathcal{L}(\theta, \mathcal{D}_{\mathcal{T}}, \mathcal{T}) + R(\theta, f_{\mathcal{T}}, \mathcal{T}, p_{\hat{\mathbf{X}}_{\mathcal{T}}}) \right]. \tag{B.27}$$

The covariance scale σ_t and diagonal offset τ_t are optimized as hyperparameters. To avoid overconfident predictions as context points approach the invariant distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we increase τ_t with the same schedule as the noise scales β_t,

starting from $\beta_0 = \tau_t$ and ending at $\beta_T = 10\tau_t$. Similarly, as the distinction between in- and out-of-distribution becomes meaningless at larger noise levels, we decrease σ_t with an inverted schedule starting from $\beta_0 = \sigma_t$ and ending at $\beta_T = 0.1\sigma_t$.

At sampling time, Gruver et al. (2023) generate sequences using a grid of denoising hyperparameters and examine the resulting Pareto front of objective values versus "naturalness" i.e. how likely a sequence is to be synthesizable, estimated by its likelihood under a protein language model (Ferruz et al., 2022). We use same hyperparameter grid (presented in Table 8), generating samples with five independently trained models with different random seeds.

Table 8: Hyperparameter search space for regularization schemes of regression models detailed in Appendix B.4.

Hyperparameter	Description	Sampling Range
λ	strength of naturalness regularization	10^1, 1, 10^{-1}, 10^{-2}, 10^{-3}
η	step size of Langevin dynamics update	1.0
K	number of steps used to update embedding	10
guidance layer	which embedding to use for update	first
return best	return samples with best guidance model predictions	true

Additional Experimental Results. In addition to the Pareto front of the generated samples shown in Figure 6, we follow Gruver et al. (2023) and also visualize the full distribution of generated samples by plotting their kernel density estimates. The results are presented in Figure 23, mirroring the conclusions drawn from the analysis of the Pareto front.

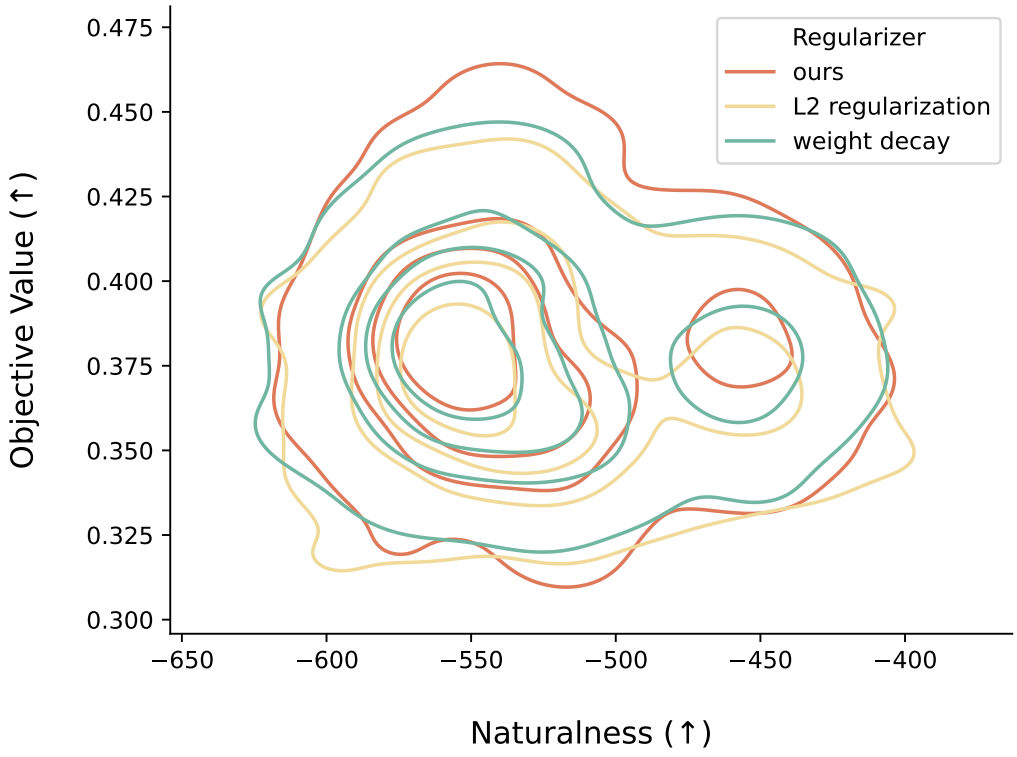


Figure 23: Kernel density estimates (KDE) of all samples generated with different regularization schemes and sampling hyperparameters, using the standard KDE parameters of the SEABORN plotting library (Waskom, 2021) and five density isocontours each. Similar to the Pareto front, this highlights the trade-off between objective value and naturalness. As samples move away from the training data and enter an out-of-distribution regime, our method consistently generates sequences with better properties at the same level of naturalness.

B.5. Runtime Comparison.

We quantify computational cost as the average execution time per mini-batch iteration, including data loading, model forward passes, backpropagation, and parameter updates. Figure 24 compares regularization schemes for the application in Section 5.1 across 5 model training runs. Figure 25 shows the same comparison for the experiments in Section 5.3. All models were trained with an identical setup and on the same NVIDIA A100 GPUs, differing only in the regularizer.

When training a standard guidance model, as in Lee et al. (2023), we observe that the added computational cost at training time is roughly equivalent to training a deep ensemble with $M = 5$ models (Figure 24). However, we note that our method does not induce any additional cost at inference, i.e., sampling time. When training the score and guidance models jointly and only updating the latter every 5 steps, as in Gruver et al. (2023), we observe that the increase in computational cost becomes negligible (Figure 25).

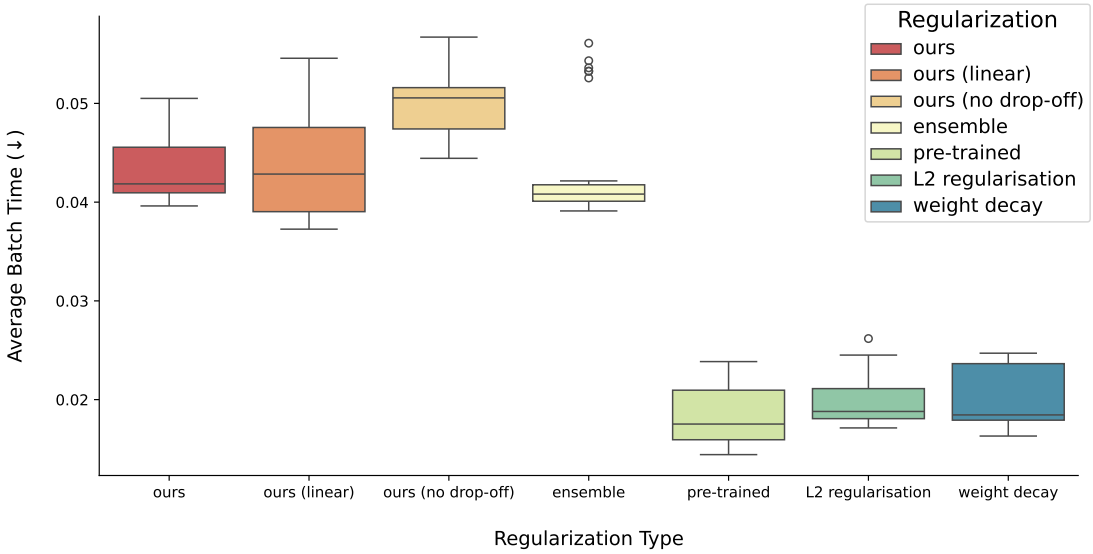


Figure 24: Average batch times of models from Section 5.1 trained with an identical setup and on the same GPUs.

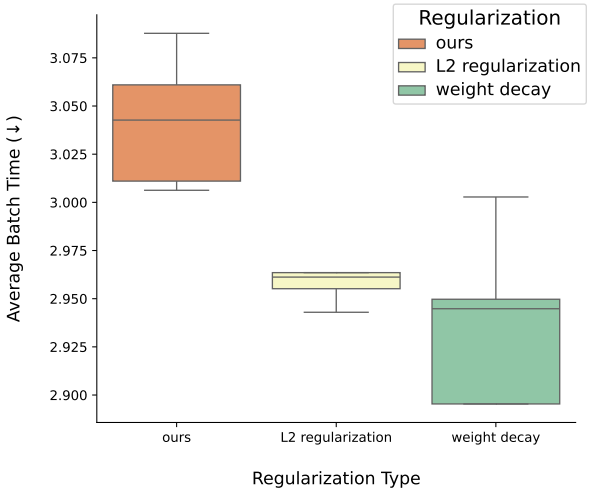


Figure 25: Average batch times of models from Section 5.3 trained with an identical setup and on the same GPUs.

B.6. Similarity of the Generated Samples to the Training and Validation Sets.

As our experimental setups select regularization hyperparameters based on an out-of-distribution validation set with more desirable properties than the training data, it is important to verify that the models do not simply generate molecules from the high-value validation set. The results are presented below and demonstrate that guidance model trained with our regularizer do not simple generate molecules from the validation set, but are instead learning to sample molecules with improved properties from novel subsets of the input domain.

B.6.1. Graph-Structured Diffusion for Small Molecules

To investigate this, we computed the maximum ECFP4-based (Rogers & Hahn, 2010) Jaccard-Tanimoto similarity of the molecules generated by our model in Section 5.1 to any compound in the training and validation sets, respectively. The results are provided in Table 9 and show that the similarities are consistently low and comparable across the training and validation sets (a threshold of 0.7 is often used to indicate molecules as closely related).

Table 9: Maximum extended-connectivity fingerprint (ECFP4; Rogers & Hahn (2010))-based Tanimoto similarity (Bajusz et al., 2015) of the small molecules generated in Section 5.1 to compounds in the training and validation sets. Means and standard deviations are reported across five independent training and sampling runs with different random seeds.

Set	PARP1	FA7	5HT1B	BRAF	JAK2
Training Set	0.33 ± 0.09	0.35 ± 0.08	0.35 ± 0.08	0.37 ± 0.07	0.36 ± 0.08
Validation Set	0.33 ± 0.09	0.36 ± 0.09	0.37 ± 0.09	0.39 ± 0.10	0.38 ± 0.09

B.6.2. Equivariant Diffusion For Materials

Similarly, we investigated the distribution of generated molecules across the training, validation, and test sets of the materials design experiments from Section 5.2. As this experimental setup operates in an exhaustively enumerated search space, we compare the proportions of generated samples across all sets with both unguided and weight decay-regularized baselines. We use a guidance scale of 4, as this is the setting in which both guided diffusion models perform best (see Figure 5). We observe that our model generates a similar proportion of molecules from the medium-value validation set as the weight decay-regularized baseline, while producing substantially more compounds from the completely held-out high-property test set.

Table 10: Distribution of generated molecules across the training, validation, and test sets for the materials science experiments in Section 5.2. Means and standard deviations are reported across ten independent training and sampling runs with different random seeds.

Model	ratio in training set	ratio in validation set	ratio in test set
Unguided	0.91 ± 0.01	0.05 ± 0.00	0.04 ± 0.00
Weight Decay	0.49 ± 0.07	0.32 ± 0.05	0.19 ± 0.03
Ours	0.19 ± 0.03	0.39 ± 0.05	0.46 ± 0.06