UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

GUIDING DIFFUSION WITH LOGICAL CONSTRAINTS: MOLECULAR GRAPH GENERATION UNDER LIPINSKI'S RULES

EMMA MENEGHINI

Thesis supervisor
SERGI ABADAL CAVALLÉ (Department of Computer Architecture)

Thesis co-supervisor
NICOLÒ NAVARIN

Degree
Master's Degree in Data Science

Master's thesis

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

To my mum and dad,
whose love and unwavering support have guided me
through the challenges of this journey and beyond.

To my beloved,
for standing by me in difficult times
with patience, strength, and kindness.

And to the passion for knowledge,
a quiet flame that has lit the way
and inspired every step of this work.

ABSTRACT

Diffusion models have become a leading approach for graph generation, offering scalability and a principled probabilistic framework. This thesis identifies and addresses a conceptual and methodological limitation in their conditional guidance: existing methods handle only simple objectives, treating complex logical constraints as black boxes. To overcome this, it introduces a probabilistic framework that embeds logical rules, expressed in disjunctive normal form, into classifier guidance. Extending DiGress, a discrete denoising model for graphs, the framework supports cardinality constraints such as Lipinski's Rule of Five. Experiments show that it effectively enforces target rules while preserving their intended semantics under mild assumptions.

Parts of this thesis were developed into the contribution *"Guided Molecular Generation through Logical Constraints"* by Emma Meneghini, Paolo Frazzetto, and Nicolò Navarin.
This work was presented at the *Special Session on Neural Networks for Graphs and Beyond*, part of the *International Conference on Artificial Neural Networks (ICANN 2025)*, Kaunas, Lithuania, and at the *AI in Complex Systems (AICS) Satellite*, part of the *Conference on Complex Systems (CCS 2025)*, Siena, Italy.

CONTENTS

1 Introduction 1
- 1.1 Background, Motivation and Overview 1
- 1.2 Research Aims 2
- 1.3 Code and Reproducibility 2
- 1.4 Outline of the Chapters 2

2 Graph Diffusion Methods for Molecular Generation 3
- 2.1 Molecular Graph Generation 3
  - 2.1.1 Drug Discovery Tasks 3
- 2.2 Graph Generative Models 4
  - 2.2.1 Graph Neural Networks 4
  - 2.2.2 Graph Transformers 5
- 2.3 Diffusion Models for Graph Generation 6
  - 2.3.1 Score-Matching with Langevin Dynamics 6
  - 2.3.2 Denoising Diffusion Probabilistic Models 7
  - 2.3.3 Score-Based Generative Models with SDEs 7
  - 2.3.4 Comparison between Diffusion Paradigms 8
- 2.4 Conditional Diffusion Models for Molecular Graphs 9
  - 2.4.1 Guidance in Molecular Diffusion Models 9
  - 2.4.2 Property Constraints in Molecular Generation 9
- 2.5 From Guidance Limitations to Logical Conditioning 10

3 DiGress: A Discrete Denoising Diffusion Model for Graphs 11
- 3.1 Notation and Problem Setting 11
- 3.2 Forward Noising Process 11
- 3.3 Inverse Noising Process 12
- 3.4 Model Training 12
- 3.5 Approximate Reverse Sampling 13
- 3.6 Model Architecture 13
  - 3.6.1 Auxiliary Features 15

4 Logical Guidance for Conditional Generation 17
- 4.1 Conditional Generation with DiGress 17
  - 4.1.1 Conditional Sampling as Variational Inference 18
  - 4.1.2 Monotonicity of Property Predictors under Guidance 19
- 4.2 Conditional Generation with DNF Guidance 20
  - 4.2.1 Cardinality Constraints in Full DNF 20
  - 4.2.2 Property Satisfaction Patterns under Cardinality Constraints 21

5 Experimental Methodology for Logical Guidance 23
- 5.1 Conditional Generation Targeting Lipinski's Rule 23
- 5.2 Baseline Guidance Mechanisms 24
  - 5.2.1 Conjunctive Guidance 24
  - 5.2.2 Black-box Rule Satisfaction Guidance 24
- 5.3 Classifiers' Architecture 24
  - 5.3.1 Architecture and Training Hyperparameters 25

5.4 The GuacaMol Dataset 25
5.4.1 Dataset Preparation and Property Statistics 25
5.5 Plan of Experiments 26
5.6 Evaluation Metrics 27
6 Results and Discussion: A Lipinski's Case Study 31
6.1 The Unconditional DiGress Model 31
6.1.1 Benchmark Performance, Diversity, and Compliance 31
6.1.2 Distributional Shift 32
6.1.3 Latent Space Analysis 32
6.2 The Impact of Guidance Strength on Guidance Rules 34
6.3 Conjunctive vs. Lipinski's Rule Guidance 34
6.3.1 Distributional Shift 36
6.3.2 Latent Space Analysis 36
6.4 Comparison between Classifier Architectures 39
6.4.1 Benchmark Performance, Diversity, and Compliance 39
6.4.2 Distributional Shift 39
6.4.3 Latent Space Analysis 42
6.5 Black-box vs. Lipinski's Rule Guidance 45
7 Conclusions 47
7.1 Future Work 47
A DNF Representation of Cardinality Constraints I
B Theoretical Analysis of Satisfaction Patterns with Cardinality Constraints III
C Additional Experimental Results VII
C.1 PCA Projections by Classifier Architecture VII
C.2 Extended Black-box Analysis VII
C.3 Illustrative Examples of Guided Molecules VII

LIST OF FIGURES

Figure 1 GNN architecture. 4
Figure 2 Graph Transformer architecture. 5
Figure 3 Sampling process of a diffusion model. 6
Figure 4 Overview of DiGress. 11
Figure 5 DiGress' Graph Transformer architecture. 14
Figure 6 Overview of the proposed logical guidance framework. 17
Figure 7 UMAP of DiGress vs. training samples. 33
Figure 8 PCA of DiGress vs. training samples. 33
Figure 9 Trade-offs in compliance, similarity, and validity (independent-2L). 35
Figure 10 UMAP of independent-2L vs. DiGress samples. 37
Figure 11 PCA of independent-2L vs. DiGress samples. 38
Figure 12 Trade-offs in compliance, similarity, and validity (shared-2L). 40
Figure 13 Trade-offs in compliance, similarity, and validity (shared-1L). 41
Figure 14 UMAP of shared-2L vs. DiGress samples. 43
Figure 15 UMAP of shared-1L vs. DiGress samples. 44
Figure 16 PCA of shared-2L vs. DiGress samples. VIII
Figure 17 PCA of shared-1L vs. DiGress samples. IX
Figure 18 Trade-offs in compliance, similarity, and validity (black-box). X
Figure 19 UMAP of black-box vs. DiGress samples. XI
Figure 20 PCA of black-box vs. DiGress samples. XII
Figure 21 Examples of molecules under Lipinski's rule guidance. XIII

1 INTRODUCTION

1.1 BACKGROUND, MOTIVATION AND OVERVIEW

Generative models are rapidly transforming how we design structured objects, like graphs, under multiple constraints [53]. Success in this setting depends not only on realism but also on controllability—the ability to steer generation toward specific goals. Diffusion models have emerged as a leading paradigm because of their stability, scalability, and strong theoretical foundations [52]. They can capture complex distributions while still allowing conditional guidance, making them powerful engines for controlled generation.

However, existing methods remain narrow: they handle only simple constraints such as thresholds or conjunctions, while richer logical rules are treated as opaque black boxes—a limitation that, to our knowledge, has not been explicitly recognised or addressed before. As a result, a broad spectrum of domain knowledge remains beyond the reach of current generative models. This limitation is particularly critical in domains where prior knowledge is inherently logical, such as drug discovery, where chemical validity and pharmacological suitability are encoded as empirical rules [2, 32]. Embedding these rules into generative models can accelerate compound discovery and reduce reliance on costly experimental screening.

This thesis takes a first step in this direction by introducing the first framework for logical guidance at scale in diffusion models. Building on DiGress [45], a discrete denoising model for graphs, we extend classifier-guided generation to incorporate logical constraints expressed in disjunctive normal form, with particular emphasis on cardinality rules. The framework bridges logical reasoning and probabilistic modelling, estimating rule satisfaction in a principled and interpretable way under minimal assumptions. In doing so, it sets a conceptual milestone: demonstrating that symbolic constraints can be embedded directly into the generative process, while rigorously preserving their intended meaning

To evaluate the framework, we turn to molecular graph generation [3]—a domain where molecules are naturally represented as graphs, with atoms as nodes and bonds as edges, allowing generative models to capture structural and chemical relationships directly. This representation makes it possible to integrate property predictors that depend on atomic connectivity, enabling property-aware molecule design. Within this setting, we focus on Lipinski's Rule of Five [27], a cornerstone of medicinal chemistry that summarises the physicochemical patterns most often associated with oral bioavailability. The rule is particularly suitable for this study because it defines a cardinality constraint (*at least three of four conditions must hold*)—a realistic yet challenging case for testing logical guidance. On the GuacaMol benchmark [4], we compare our logical framework with conjunctive and black-box baselines, assessing compliance, validity, and distributional composition. The experiments show that our method not only enforces the rule but also reshapes the distribution of satisfying patterns, preserving the logical structure that the baselines fail to capture.

In this way, the thesis positions logical guidance as a transparent and controllable alternative to existing conditional methods. By embedding symbolic rules into the probabilistic structure of diffusion models, it contributes to the broader effort of probabilistic–symbolic integration in generative modelling. This approach points toward a new generation of models that are both

powerful and interpretable, with potential applications in molecular discovery, materials design, and other domains where structured data must satisfy complex logical constraints.

1.2 RESEARCH AIMS

The work presented in this thesis encompasses all stages of a research project, from problem formulation to theoretical development, experimentation, and analysis. Its aims were to:

- *Identify the research gap and select the base model.* Recognise the absence of principled methods for handling logical constraints within existing conditional diffusion frameworks and determine DiGress as a suitable foundation for extending guidance;
- *Define the task and benchmark.* Formulate property-based conditional generation as a suitable task in molecular graph generation, using Lipinski's Rule of Five as a representative logical constraint, and select GuacaMol as a benchmark reflecting realistic drug-like chemical spaces;
- *Develop the logical guidance framework.* Design the method both conceptually and theoretically, establishing its probabilistic foundations and interpretability properties;
- *Implement and evaluate the approach.* Define baselines, design the complete experimental and evaluation pipeline, and perform the necessary experiments; and
- *Analyse results and limitations.* Assess the framework's performance, identify its limitations, and outline directions for future research.

1.3 CODE AND REPRODUCIBILITY

To support transparency and reuse, the complete implementation of this thesis is available in the DiGress-logical-guidance GitHub repository. It combines adapted components from DiGress with newly developed modules and evaluation scripts, enabling both reproducibility of results and extension of the framework.

1.4 OUTLINE OF THE CHAPTERS

The remainder of this thesis is structured as follows:

- Chapter 2 surveys diffusion graph generative models for molecular generation, emphasising current limitations in conditional guidance;
- Chapter 3 introduces DiGress, which serves as the foundation for the proposed extension;
- Chapter 4 presents the proposed framework for logical guidance, its probabilistic formulation for cardinality rules, and the theoretical results ensuring interpretability;
- Chapter 5 defines the experimental setup, covering Lipinski's Rule of Five, baselines, and evaluation methodology;
- Chapter 6 reports and discusses the experimental results obtained on the GuacaMol benchmark;
- Chapter 7 concludes the thesis, summarising the main contributions, discussing limitations and outlining future research directions.

2

GRAPH DIFFUSION METHODS FOR MOLECULAR GENERATION

In this chapter, we review the state of the art in deep graph generative models, including conditional guidance mechanisms, with a particular focus on diffusion models. We also examine the most relevant techniques for *de novo* molecule generation, a prominent application area for these methods. Throughout the chapter, we aim to highlight key research gaps—both methodological and application-specific—and to identify the works that have most directly informed the development of our methodology.

2.1 MOLECULAR GRAPH GENERATION

Graph generation refers to the task of synthesising realistic graph-structured data that reflect key characteristics observed in real-world graphs, including both structural and domain-specific properties [17]. This task presents several challenges, such as *permutation invariance*, since graph identity should not depend on node ordering; *sparsity*, as many real-world graphs are far from fully connected; the need to model complex dependencies among nodes and edges; and *scalability*, as generating or learning from large graphs often imposes significant computational constraints. The development of graph generative models is motivated by a wide range of application domains, with molecule design representing a particularly prominent example [43].

Molecules are modelled as undirected, heterogeneous, and static graphs because chemical bonds are mutual (undirected), involve different atom and bond types (heterogeneous), and the structure remains fixed during analysis (static). In addition, there exists an alternative representation known as SMILES strings, which encodes molecules as linear sequences of characters that describe their structure using a depth-first traversal of the molecular graph [47, 49, 48].

2.1.1 *Drug Discovery Tasks*

In *de novo* drug design the goal is to generate novel chemical compounds with desirable pharmacological and physicochemical properties. Modelling molecules as graphs has enabled recent advances in generative modelling, making this process more efficient and better integrated into modern drug discovery workflows.

Molecular generative models operate either *unconditionally*, by sampling from a learned distribution, or *conditionally*, by guiding the generation toward specific properties, structural motifs, or biological targets. Conditional generation encompasses a variety of strategies—including ligand-based and structure-based design, conformer generation and molecular optimisation—that support different stages of the drug discovery process. For an in-depth overview of these tasks and their respective roles in computational molecular design, the reader is referred to a recent survey [46].

In this work, we focus specifically on *property-based conditional generation*, where the goal is to bias the generative process toward molecules that satisfy desired chemical properties or rules.

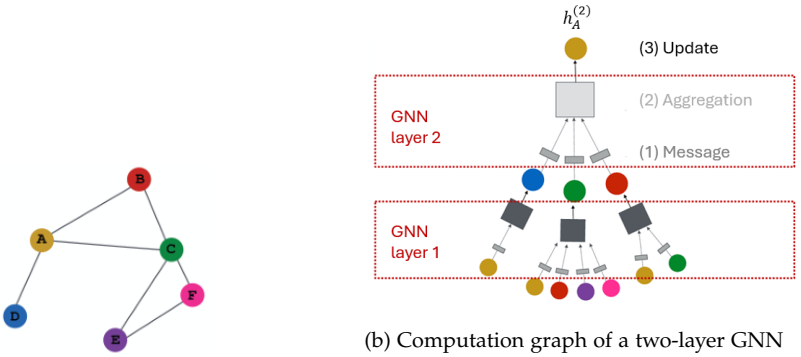
This task is particularly relevant in early-stage screening and lead optimisation, and forms the basis of the following chapters.

2.2 GRAPH GENERATIVE MODELS

Deep generative models for graphs aim to learn the underlying distribution of graph-structured data directly from examples. They can be broadly grouped into two classes: one-shot models, such as variational autoencoders [23, 41] and Generative Adversarial Networks [9], which generate an entire graph in a single pass; and autoregressive models, which build graphs sequentially, typically node by node [26]. Normalising flow models can belong to either class, depending on their formulation [30, 10]. However, none of these approaches fully address the key challenges of graph generation: permutation invariance, discrete and sparse structures, and scalability without mode collapse. In contrast, diffusion models overcome these challenges and combine stable training, typical of autoregressive models, with the broad distributional coverage of one-shot methods.

All of the generative models typically rely on Machine Learning (ML) architectures—most commonly Graph Neural Networks (GNNs)—to encode structural information and capture node and edge dependencies during generation.

2.2.1 *Graph Neural Networks*



(a) Example of an undirected input graph for a GNN, with input node A.

(b) Computation graph of a two-layer GNN for node A, where messages from each neighbour (and A itself) are aggregated and updated layer by layer.

Figure 1: GNN architecture as in [25].

While ML models often rely on learned rather than hand-crafted features, their performance is still shaped by the inductive biases built into their architectures. In the case of graph-structured data, GNNs offer a principled way to incorporate relational inductive biases by exploiting the structure of the underlying graph. They learn by repeatedly exchanging information between a node and its neighbours through a process known as *message passing* (MP). The number of layers in a GNN determines how far information can travel through the graph, as each layer allows messages to propagate one step further.

There are many variants of GNNs [22], but they all share a common framework, illustrated in Figure 1, in which the representation of a node u at layer l is updated according to:

$$h_u^{(l)} = \text{UPDATE}^{(l)}\left(m_u^{(l)},\ \text{AGGREGATE}^{(l)}\left(\left\{ m_v^{(l)} \mid v \in \mathcal{N}(u)\right\}\right)\right), \tag{2.1}$$

$$m_v^{(l)} = \text{MSG}\left(h_v^{(l-1)}\right) \quad v \in \{\mathcal{N}(u) \cup u\}, \tag{2.2}$$

where $m_v^{(l)}$ is the message of node v, defined as a (non-)linear transformation MSG of its representation $h_v^{l-1} \in \mathbb{R}^{d_{l-1}}$, and $\mathcal{N}(u)$ denotes the set of neighbours of node u. The AGGREGATE function is a symmetric operator (e.g., sum, mean, maximum), and UPDATE is a differentiable function, typically a neural network. The same aggregation and update parameters are shared across all nodes, meaning that each node applies the same learnable functions to its local neighbourhood, which makes the model independent of the specific graph size or node identities. Moreover, the symmetry of the aggregation function guarantees invariance to the order of the neighbours, thereby preserving permutation equivariance. As we will see later, the equivariance of the architecture is one of two fundamental components underpinning the equivariance properties of a graph generative model.

Despite its effectiveness, the sequential MP scheme of Equation (2.1) suffers from several well-known bottlenecks. First, information must propagate layer by layer, making long-range interactions computationally expensive. This results in *under-reaching*, where relevant signals from distant nodes fail to influence the representation of a given node. This issue is exacerbated in small, sparse graphs such as molecules, where the graph diameter can be large relative to the number of nodes, making long-range dependencies particularly important. Second, GNNs tend to *over-smooth*, causing node representations to become indistinguishable. Finally, they often *over-squash*, limiting their capacity to encode information from large neighbourhoods.

Next, we introduce Graph Transformers, which mitigate these limitations by generalising the MP mechanism of standard GNNs.

2.2.2 *Graph Transformers*

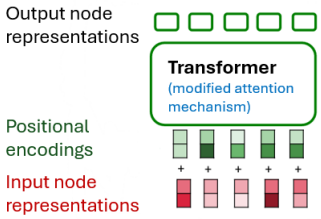


Figure 2: Graph Transformer architecture as in [25].

Graph Transformers replace the local aggregation typical of MP-GNNs with a fundamentally different mechanism: global *self-attention* [12]. This architecture has gained popularity in the molecular ML literature, including property prediction [19, 31] and generative tasks [45, 51, 42, 29], where capturing interactions between atoms that are not directly connected is essential for modelling chemically realistic structures.

Multi-head self-attention computes weighted combinations of features across the entire graph, allowing nodes to access long-range information without deep architectures, as illustrated

in Figure 2. To maintain awareness of graph structure, structural and positional encodings are integrated into the attention mechanism. In molecular graphs, edge-aware bias terms can further enrich attention scores, preserving the heterogeneity of atomic interactions. For improved scalability, some variants adopt local–global hybrid attention, restricting global attention to a subset of nodes.

Overall, Graph Transformers alleviate key GNN bottlenecks: they address under-reaching by enabling efficient modelling of long-range dependencies, reduce over-smoothing by avoiding excessive depth, and mitigate over-squashing through direct global information flow [33].

Although GNNs provide strong graph representations, generative modelling requires learning full data distributions. We now review diffusion models that extend this capability.

2.3 DIFFUSION MODELS FOR GRAPH GENERATION

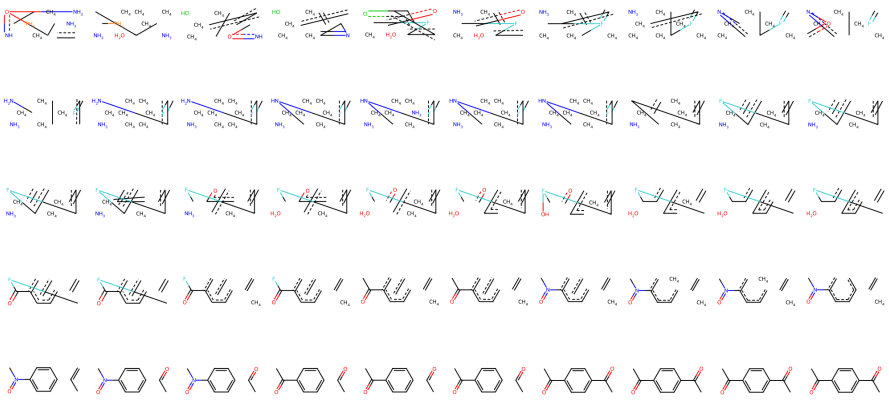


Figure 3: Sampling process of a D3PM for molecular graphs, starting from pure noise (top left) and converging to a fully denoised molecule (bottom right).

Diffusion models generate data by learning to reverse a gradual noising process, as illustrated in Figure 3. Diffusion consists of a *forward* corruption step and a *reverse denoising* step, parameterised by a neural network (the model). Different formulations vary in the design of the forward and reverse processes, as well as in training and sampling methods—affecting flexibility, efficiency, and applicability to structured domains like graphs.

The following sections introduce score-based models with discrete noise levels, followed by their extension to continuous-time formulations via stochastic differential equations (SDEs), which unify prior approaches and enable more efficient sampling. Finally, the approaches are compared.

2.3.1 *Score-Matching with Langevin Dynamics*

In score-based diffusion models with Langevin Dynamics (SMLD) [35], the goal is to learn a function that estimates the *score*, which is the gradient of the log-density $\nabla_G \log q(G)$ and points toward regions of higher data likelihood in the input space. Since the true data distribution $q(G)$

is intractable, the model instead learns to approximate the score of a noisy version G_σ, obtained by perturbing a clean graph G with Gaussian noise at varying noise levels σ.

From a Bayesian perspective, the score can be expressed using Tweedie's formula:

$$\nabla_{G_\sigma} \log q_\sigma(G_\sigma) = \frac{1}{\sigma^2} \left(\mathbb{E}[G \mid G_\sigma] - G_\sigma\right), \tag{2.3}$$

which shows that estimating the posterior mean $\mathbb{E}[G \mid G_\sigma]$ is equivalent to learning the score function. In practice, a neural network $s_\theta(G_\sigma, \sigma)$ is trained to approximate this quantity across a range of noise levels via a denoising score-matching loss.

Although SMLD avoids explicit likelihood estimation, it typically requires many Langevin sampling steps with small step sizes. Moreover, since it operates on continuous relaxations of graph data, an additional discretisation step is needed to recover valid graphs.

2.3.2 *Denoising Diffusion Probabilistic Models*

Denoising diffusion probabilistic models (DDPMs) define a forward diffusion process that gradually corrupts a clean graph G^0 through a Markov Chain of Gaussian transitions. In models such as ConGress [45], noise is added independently to node features X and edge features E at each step according to a schedule α_t. Because the Gaussian transitions yield a closed-form marginal, the distribution at any step t has a closed form conditioned on the original graph:

$$q\left(X^t \mid X\right) = \mathcal{N}\left(X^t \mid \alpha^t X,\, \sigma^t I\right), \tag{2.4}$$

$$q\left(E^t \mid E\right) = \mathcal{N}\left(E^t \mid \alpha^t E,\, \sigma^t I\right), \tag{2.5}$$

where σ^t is determined by the value of α^t. This formulation enables efficient training by allowing direct sampling from intermediate states without simulating the entire forward chain.

To reverse the corruption process, DDPMs learn a parameterised reverse Markov Chain $q_\theta\left(G^{0:T}\right)$ by optimising a variational evidence lower bound (ELBO) on the log-likelihood of G^0. This reduces to a denoising loss, where the model is trained to predict the added noise ϵ from a noisy input G^t and time step t. Since $\nabla_{G^t} \log q\left(G^t\right) \propto -\epsilon$, this corresponds to estimating the score function up to a known scale—making DDPM training equivalent to denoising score matching as used in SMLD.

DDPMs share the same limitations as SMLD, requiring many sequential sampling steps and relying on continuous data representations, which complicates the modelling of inherently discrete structures such as molecular graphs.

2.3.3 *Score-Based Generative Models with SDEs*

A more general and flexible framework is provided by score-based generative models with stochastic differential equations (SGMs)—such as GDSS [21]—which model the diffusion process as a continuous-time stochastic differential equation:

$$\mathrm{d}G^t = f\left(t, G^t\right)\mathrm{d}t + g(t)\mathrm{d}W^t, \quad G^0 \sim p_{data}, \tag{2.6}$$

where W^t denotes a standard Wiener process. The time-dependent score function $\nabla_{G^t} \log q\left(G^t\right)$ is learned via a denoising score-matching objective equivalent in principle to those used in SMLD and DDPMs. Once learned, it defines the reverse-time SDE used for sampling.

This continuous-time formulation unifies and generalises both methods: it recovers DDPMs in the discrete time limit with variance-preserving noise, corresponds to SMLD when sampling is performed via Langevin dynamics at fixed noise levels, and further enables flexible noise schedules and adaptive numerical integration for more efficient and accurate sampling.

Importantly, the framework also admits a deterministic alternative to the reverse SDE, known as the probability flow ordinary differential equation (ODE). As a reverse process, it can reduce sampling variance and inference steps, but requires accurate score estimates, well-tuned noise schedules, and stable ODE solvers, which may be costly in high-dimensional settings.

2.3.4 *Comparison between Diffusion Paradigms*

While score-based generative models based on SDEs provide a unified theoretical framework that encompasses both continuous and discrete-time models, there is currently no definitive evidence that one formulation consistently outperforms the other empirically—particularly in the context of graph data. Regarding the advantages of using a discrete-time model, [24] observe that such models are generally more tractable and easier to implement than their continuous-time counterparts. Each step in the forward and reverse processes involves functionally defined transition distributions, enabling straightforward iterative sampling without the need for numerical solvers. While continuous-time models benefit from stronger theoretical tools via stochastic calculus, discrete-time approaches typically incur lower computational cost and are more practical for real-world applications.

When comparing continuous and discrete state spaces for graph generation, [16] observe that discrete-state models maintain graph sparsity by avoiding Gaussian noise, thereby achieving lower Maximum Mean Discrepancy (MMD) with respect to the test set, requiring shorter diffusion chains, and enabling faster sampling.

Recent work has aimed to unify discrete and continuous formulations across both time and state space [5, 39]. For the first time, [51] introduced a discrete-state, continuous-time diffusion model for graphs, aiming to decouple the number of sampling steps from the diffusion steps used during training. [42] developed a novel discrete-time, continuous-state diffusion model that incorporates continuous graph topological information into the generative process. This design promotes smooth and consistent information flow during generation, avoiding the abrupt structural changes often caused by discrete transitions. As a result, the model achieves more efficient sampling and demonstrates competitive performance with baseline methods.

Both [51] and [42] compare their methods directly with DiGress [45], a discrete-time, discrete-state diffusion model (D3PM), with [51] even adopting the same diffusion architecture. Both works acknowledge DiGress for its originality, scalability, and consistent performance across planar, community-based, and molecular benchmarks. Given the similarity in reported performance, DiGress remains a strong and well-established baseline, offering a simpler and empirically robust alternative.

While these models offer powerful generative capabilities, most practical applications—particularly in drug discovery—require controlled generation conditioned on specific molecular properties. We now turn to the conditional guidance mechanisms that address this need.

2.4 CONDITIONAL DIFFUSION MODELS FOR MOLECULAR GRAPHS

This section outlines the principal conditional paradigms for diffusion models together with the property constraints commonly applied in drug discovery.

2.4.1 *Guidance in Molecular Diffusion Models*

Conditional guidance in diffusion models for graphs has evolved into three main paradigms: classifier-based guidance, classifier-free guidance, and guidance interpreted through the lens of stochastic optimal control. These approaches aim to steer the generative process toward molecules that satisfy specific target property values, enabling controlled molecular design. In addition to these, projection-based methods have also been proposed [40], but their applicability remains limited due to the difficulty of defining suitable projection operators for complex constraints.

CLASSIFIER GUIDANCE. The constraints are learned by a predictor trained independently alongside the diffusion model, and its gradients are injected back into the reverse step to bias generation. Consequently, this guidance mechanism allows the same unconditional model to be reused with different property predictors. For example, [45] inject gradients from a frozen property predictor into the reverse process of DiGress. [20] similarly guide the CDGS reverse-time ODE using gradients from a predictor.

CLASSIFIER-FREE GUIDANCE. It avoids the need to train auxiliary predictors by jointly training conditional and unconditional diffusion models, and by combining their score estimates at sampling time. For example, [34] adapt this approach to DiGress, highlighting that classifier-based guidance relies on strong distributional assumptions that are often violated in practice. A similar strategy is employed by Graph DiT [29], which integrates conditioning directly into the generative model. This method can also accommodate non-differentiable constraints—i.e., constraints that cannot be learned by a predictor—since it does not rely on gradient computations.

STOCHASTIC OPTIMAL CONTROL. It offers a flexible alternative to the former two approaches by formulating conditional generation as a control problem. GGDiff exemplifies this approach by introducing a unified framework that supports both differentiable and non-differentiable reward functions [44]. It injects a control signal into the reverse diffusion process, using gradient-based guidance when rewards are differentiable, and zeroth-order optimisation otherwise. This allows GGDiff to guide pre-trained models without retraining, enabling effective conditioning under complex constraints.

2.4.2 *Property Constraints in Molecular Generation*

Diffusion models use conditional mechanisms to guide molecule generation toward desirable characteristics [3, 28, 43, 46]. These constraints are often incorporated into rule-based filters [32, 2], which can be modelled as general logical rules—disjunctions of conjunctions of binary conditions on molecular descriptors. In practice, however, many of these rules exhibit a simple mathematical structure, often requiring that all or a subset of the conditions be satisfied

simultaneously; we refer to these as *cardinality rules*. The most prominent among these is Lipinski's Rule of Five [27]. This guideline states that a compound is more likely to be orally active if *at least* three of four criteria are met, relating to molecular weight, lipophilicity, and hydrogen bonding capacity—favouring small, modestly lipophilic molecules with limited numbers of hydrogen bond donors and acceptors.

2.5 FROM GUIDANCE LIMITATIONS TO LOGICAL CONDITIONING

Although the frameworks reviewed in Section 2.4.1 handle both differentiable and non-differentiable objectives effectively, they leave unaddressed an orthogonal problem: modelling logical constraints in a way that preserves their intended semantics through a formal mathematical formulation. This limitation restricts their applicability, particularly in molecular design, where many domain rules exhibit a non-trivial logical structure. Such cases include cardinality rules, such as Lipinski's Rule of Five [27].

On the one hand, standard classifier guidance is inadequate: previous attempts to model the logical structure of the rule have been non-probabilistic [45], introducing bias for rules other than pure conjunctions. Consequently, it can only learn the rule as a binary target, treating it as a black-box objective and discarding supervision from individual properties. As shown in our experiments (Chapter 6), this often leads to *shortcut learning*, where the model formally satisfies the rule but neglects parts of the satisfying set. On the other hand, classifier-free guidance is rigid, as it requires access to an exact constraint vector during both training and inference, making it unsuitable for rules that allow partial satisfaction. Finally, zeroth-order methods are computationally expensive and prone to bias, limiting their practical usability.

Moreover, existing conditional diffusion models are rarely applied to large-scale generation under filters that define broad regions of the molecular space. Instead, they are typically used to generate a small number of molecules with specific target property values [45, 20, 34].

These limitations reveal a gap between current guidance paradigms and the requirements for modelling general logical constraints, both within molecular design and beyond. To bridge this gap, this thesis introduces a generalisation of classifier-based guidance through a principled mathematical formulation that embeds such constraints directly into the generative process, providing per-property supervision to enhance interpretability and preserve rule structure. Tractability and scalability are ensured by leveraging DiGress [45] as a strong baseline, while keeping additional assumptions minimal. The proposed method is evaluated on Lipinski's Rule of Five, which exemplifies this class of logical constraints and serves as a prominent benchmark in drug design.

3 DIGRESS: A DISCRETE DENOISING DIFFUSION MODEL FOR GRAPHS

In this chapter, we detail the functioning of DiGress [45], including its unconditional generative process and architecture. This discussion provides the necessary foundations for the following chapter, where we introduce its original guidance mechanism and present our proposed extension for conditional generation.

3.1 NOTATION AND PROBLEM SETTING

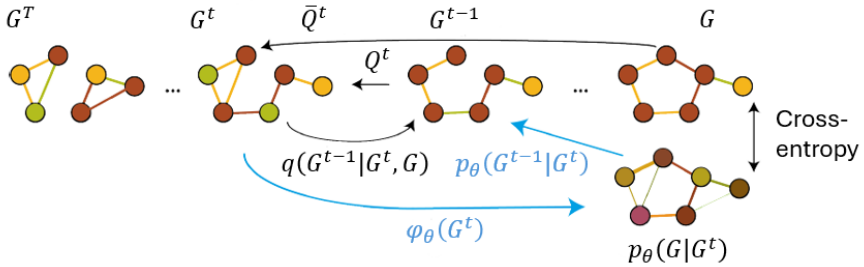


Figure 4: Overview of DiGress. The noise model is defined by Markov transition matrices Q^t whose cumulative product is $\overline{Q}^t$. The denoising network φ_θ learns to predict the clean graph from G^t. During inference, the predicted distribution is combined with $q\left(G^{t-1}|G^t, G\right)$ in order to compute $p_\theta\left(G^{t-1}|G^t\right)$ and sample a discrete G^{t-1} from this product of categorical distributions.

DiGress is a graph diffusion model, specifically a D3PM, designed to handle categorical node and edge features, which take values in finite sets $\mathcal{X}$ and $\mathcal{E}$, of cardinalities a and b, respectively. A graph is denoted by $G = (X, E)$, where $X \in \mathbb{R}^{n \times a}$ is the one-hot encoded matrix of node attributes, and $E \in \mathbb{R}^{n \times n \times b}$ is the one-hot encoded tensor of edge attributes—representing, in the molecular generation context, atom and bond types. The absence of an edge is represented as a dedicated category within $\mathcal{E}$. An overview of the forward and reverse diffusion processes is shown in Figure 4.

3.2 FORWARD NOISING PROCESS

The diffusion process unfolds over T discrete time steps, governed by Markov transition matrices $\{Q_X^t\}_{t=1}^T$ and $\{Q_E^t\}_{t=1}^T$, and converges to a stationary distribution that matches the empirical marginals of the training data. At each step t, $Q_X^t \in \mathbb{R}^{a \times a}$ and $Q_E^t \in \mathbb{R}^{b \times b}$ encode the categorical transition probabilities for each node and edge feature, applied independently to each element:

$$[Q_X^t]_{ij} = q\left(x^t = j \,|\, x^{t-1} = i\right) \quad \text{and} \quad [Q_E^t]_{ij} = q\left(e^t = j \,|\, e^{t-1} = i\right). \tag{3.1}$$

It follows that the forward kernel takes the form:

$$q\left(G^t \mid G^{t-1}\right) = \left(X^{t-1}Q_X^t,\, E^{t-1}Q_E^t\right) \quad \text{and} \quad q\left(G^t \mid G\right) = \left(X\overline{Q}_X^t,\, E\overline{Q}_E^t\right), \tag{3.2}$$

where $\overline{Q}_X^t = Q_X^1 \cdots Q_X^t$ and $\overline{Q}_E^t = Q_E^1 \cdots Q_E^t$. Since molecular graphs are undirected, noise is applied only to the upper–triangular part of E, and the tensor is symmetrised afterwards.

3.3 INVERSE NOISING PROCESS

Due to the discrete, Markovian nature of the forward kernel in Equation (3.2), the exact posterior over the previous noisy graph G^{t-1} given the current corrupted graph G^t and the original graph G admits a closed-form expression:

$$q\left(G^{t-1} \mid G^t, G\right) \propto \left(X^t Q_X^{t\top} \odot X\,\overline{Q}_X^{t-1},\, E^t Q_E^{t\top} \odot E\,\overline{Q}_E^{t-1}\right), \tag{3.3}$$

where $\odot$ denotes the element-wise (Hadamard) product. Equation (3.3) can be read component-wise:

$$q\left(x^{t-1} = i \mid X^t, X\right) \propto \left(X^t Q_X^{t\top}\right)_i \left(X\,\overline{Q}_X^{t-1}\right)_i, \tag{3.4}$$

$$q\left(e^{t-1} = i \mid E^t, E\right) \propto \left(E^t Q_E^{t\top}\right)_i \left(E\,\overline{Q}_E^{t-1}\right)_i, \tag{3.5}$$

so the posterior for each node or edge category is obtained by multiplying (1) the likelihood of observing the current noisy category after one step, given by transition kernels Q_X^t and Q_E^t, with (2) the prior probability of that category under the $(t-1)$-step kernels $\overline{Q}_X^{t-1}$ and $\overline{Q}_E^{t-1}$.

3.4 MODEL TRAINING

DiGress consists of a denoising network φ_θ, trained by maximising the ELBO on the log-likelihood of G, enabling it to reconstruct clean graphs from their progressively noised counterparts. Let $\hat{p}_{i,\theta}^X(x) = \mathbb{P}_\theta\left(x_i = x | G^t\right)$ denote the model's predicted probability that the i-th node in the clean graph takes value x, given the current noisy graph G^t. Because for nodes—and similarly for edges—the learned distribution is expressed as the product of the exact inverse distribution q and an approximate component $\hat{p}_\theta$:

$$p_\theta\left(x_i^{t-1} \mid G^t\right) = \sum_{x \in \mathcal{X}} q\left(x_i^{t-1} \mid x_i^t,\, x_i = x\right) \hat{p}_{i,\theta}^X(x) \quad \text{if} \quad q\left(x_i^t | x_i = x\right) > 0, \tag{3.6}$$

the training objective specialises into a weighted sum of cross-entropy losses:

$$\mathcal{L}(G, \hat{p}_{G,\theta}) = \sum_i \mathrm{CE}\left(x_i, \hat{p}_{i,\theta}^X\right) + \gamma \sum_{i,j} \mathrm{CE}\left(e_{ij}, \hat{p}_{ij,\theta}^E\right), \tag{3.7}$$

where γ controls the relative importance of edge terms. Since DiGress operates consistently on isomorphic graphs without requiring explicit canonicalisation (see the equivariance properties in [45]), its likelihood estimates are well-defined and meaningful. As a result, training remains computationally efficient and stable, as it leverages standard classification losses while still capturing complex graph structures. The complete training procedure is presented in Algorithm 3.1.

Algorithm 3.1 Training DiGress

Input: A graph $G = (X, E)$
Sample $t \sim \mathcal{U}\{1, \dots, T\}$
Sample $G^t \sim X\overline{Q}_X^t \times E\overline{Q}_E^t$ ▷ Sample a (discrete) noisy graph
$z \leftarrow f(G^t, t)$ ▷ Structural and spectral features
$(\hat{p}^X, \hat{p}^E) \leftarrow \varphi_\theta(G^t, z)$ ▷ Forward pass
Update θ using loss (3.7) ▷ Cross-entropy

3.5 APPROXIMATE REVERSE SAMPLING

After training, the denoising network φ_θ is used to define an approximate reverse kernel $p_\theta(G^{t-1} \mid G^t)$, which enables sampling from the model by iteratively denoising a fully noised graph. The sampling process is formalised in Algorithm 3.2.

Algorithm 3.2 (Unconditional) Sampling from DiGress

Sample n from the training data distribution
Sample $G^T \sim q_X(n) \times q_E(n)$ ▷ Random graph
for $t = T, \dots, 1$ **do**
  $z \leftarrow f(G^t, t)$ ▷ Structural and spectral features
  $(\hat{p}^X, \hat{p}^E) \leftarrow \varphi_\theta(G^t, z)$ ▷ Forward pass
  Compute $p_\theta\left(x_i^{t-1} \mid G^t\right)$ and $p_\theta\left(e_{ij}^{t-1} \mid G^t\right)$ using (3.6)
  Sample $G^{t-1} \mid G^t \sim \prod_{1 \le i \le n} p_\theta\left(x_i^{t-1} \mid G^t\right) \prod_{1 \le i,j \le n} p_\theta\left(e_{ij}^{t-1} \mid G^t\right)$
  ▷ Categorical distribution
end for
return G^0

3.6 MODEL ARCHITECTURE

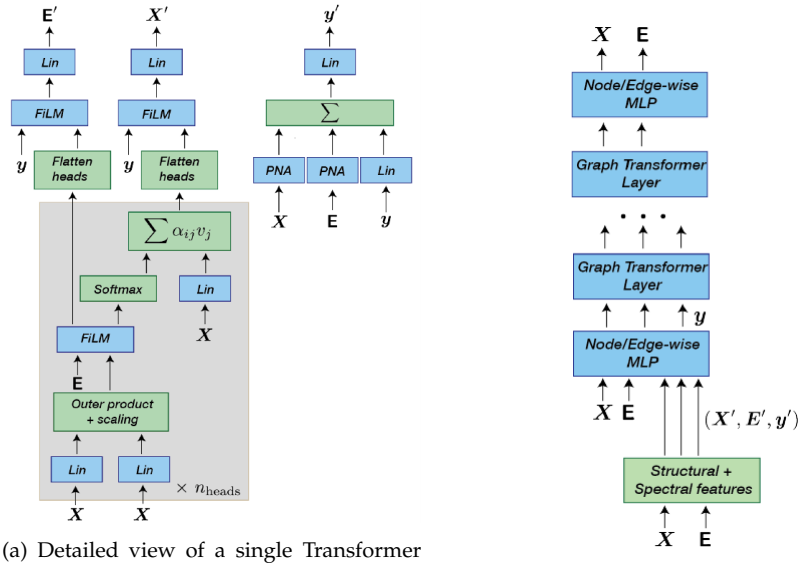
DiGress employs a Graph Transformer architecture, which is a stack of alternating blocks of multi-head attention (Attn) and Feed-Forward Networks (FFN). In each layer $l > 0$, the input features $X^{(l)} \in \mathbb{R}^{n \times d}$ are projected into *queries*, *keys*, and *values*:

$$Q = X^{(l)} W_Q, \quad K = X^{(l)} W_K, \quad V = X^{(l)} W_V, \tag{3.8}$$

where $W_Q, W_K \in \mathbb{R}^{d \times d_k}$, and $W_V \in \mathbb{R}^{d \times d}$ are learnable weight matrices. A single-head attention operation is then defined as:

$$\mathrm{Attn}\left(X^{(l)}\right) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \tag{3.9}$$

The softmax function is applied row-wise to produce attention weights, allowing each node to attend to all others in the graph and enabling the model to capture long-range dependencies.



(a) Detailed view of a single Transformer layer.

(b) High-level pipeline.

Figure 5: Graph Transformer architecture as in [45].

In practice, multi-head attention is used, where multiple such heads are computed in parallel and their outputs concatenated. Finally, an FFN is applied, completing the Transformer layer:

$$X^{(l+1)} = \mathrm{FFN}\Big(\mathrm{MultiHead}(X^{(l)}) + X^{(l)}\Big). \tag{3.10}$$

DiGress extends the standard architecture design by incorporating rich edge-aware and graph-level information through Feature-Wise Linear Modulation (FiLM) conditioning [36] and global context vectors, as shown in Figure 5. While conventional Graph Transformers incorporate structure via positional encodings added as bias terms to the attention logits, this model instead uses edge embeddings to modulate the attention scores directly. Each edge feature is projected into a multiplicative term E_{mul} and an additive term E_{add}, which are combined with the logits through a FiLM-like transformation:

$$\mathrm{Attn}\Big(X^{(l)}\Big) = \mathrm{softmax}\left(\left(\frac{QK^\top}{\sqrt{d_k}} \cdot \Big(E^{(l)}_{\text{mul}} + 1\Big)\right) + E^{(l)}_{\text{add}}\right) V. \tag{3.11}$$

In addition to node and edge features, the model maintains a global graph-level feature vector y, updated independently at each layer using Principal Neighbourhood Aggregation (PNA) summaries of node and edge states. These summaries are combined with a transformed version of the previous global vector through summation and a linear projection. The resulting global representation then modulates node and edge representations via FiLM layers, enabling global context to influence the update dynamics across the graph.

Overall, this design enables bidirectional information flow between local and global levels of the graph, enhancing the model's ability to reason over complex, long-range dependencies more effectively than standard Graph Transformer variants.

3.6.1 *Auxiliary Features*

To enhance its architectural expressiveness, DiGress incorporates additional structural features not captured by standard attention-based processing. These are grouped into two categories: graph-theoretic and domain-specific. The former includes cycle counts—added at both node and graph level for cycles up to size 6—and spectral descriptors such as the number of connected components, the smallest non-zero Laplacian eigenvalues, and selected node-level eigenvector components. These augmentations help the model capture both global and local structural patterns. The latter comprises chemical features such as atomic valency and overall molecular weight, which improve performance on molecular datasets. Although optional, these features enhance generation quality and are efficiently computable at each diffusion step.

4 LOGICAL GUIDANCE FOR CONDITIONAL GENERATION

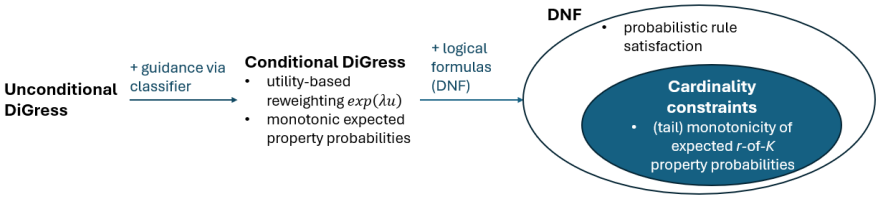


Figure 6: DiGress is extended with classifier guidance that models the probabilistic satisfaction of DNF formulas, with cardinality constraints forming a tractable subclass. We derive the properties of both the original conditional DiGress model [45] and our proposed guidance extension.

In this chapter, we first present the conditional guidance framework introduced in DiGress [45] and derive some of its properties. We then extend it to handle logical constraints—particularly cardinality rules—in a probabilistic manner, and provide a theoretical analysis of the resulting formulation. A schematic overview of the proposed logical guidance mechanism and its main theoretical components is shown in Figure 6.

4.1 CONDITIONAL GENERATION WITH DIGRESS

The DiGress conditional framework separates the unconditional denoising process from the estimation of the probability that a generated graph satisfies the desired properties, which is computed by a classifier or regressor. This decoupling allows the same diffusion model to be reused for different targets, making conditional generation flexible and computationally efficient. The conditional denoising process in DiGress is formalised in the following lemma.

Lemma 4.1. *(Conditional denoising process) [45, 11] Let q and $\dot{q}$ denote the unconditional and conditional noising processes with respect to y_G, respectively, and assume that $\dot{q}(G^t \mid G, y_G) = \dot{q}(G^t \mid G)$. Then we have $\dot{q}(G^{t-1} \mid G^t, y_G) \propto q(G^{t-1} \mid G^t)\, \dot{q}(y_G \mid G^{t-1})$.*

We would like to estimate $q(G^{t-1} \mid G^t)\, \dot{q}(y_G \mid G^{t-1})$ by leveraging its parameterised counterparts, namely the diffusion model $p_\theta(G^{t-1} \mid G^t)$ and the predictor $p_\eta(y_G \mid G^{t-1})$, which together approximate the unconditional reverse kernel and the conditional property likelihood, respectively. However, p_η does not factorise as a product over nodes and edges and cannot be evaluated for all possible values of G^{t-1}. To overcome this issue, it is sufficient to view G

as a continuous tensor of order $n + n^2$ (so that ∇_G can be defined), and use a first-order approximation. This gives:

$$\log \dot{q}\left(y_G \mid G^{t-1}\right) \approx \log \dot{q}\left(y_G \mid G^t\right) + \left\langle \nabla_G \log \dot{q}\left(y_G \mid G^t\right),\, G^{t-1} - G^t \right\rangle \tag{4.1}$$

$$\approx c\left(G^t\right) + \sum_{1 \le i \le n} \left\langle \nabla_{x_i} \log \dot{q}\left(y_G \mid G^t\right),\, x_i^{t-1} \right\rangle + \sum_{1 \le i,j \le n} \left\langle \nabla_{e_{ij}} \log \dot{q}\left(y_G \mid G^t\right),\, e_{ij}^{t-1} \right\rangle, \tag{4.2}$$

for a function c that does not depend on G^{t-1}. The resulting conditional sampling procedure is presented in Algorithm 4.1.

Algorithm 4.1 Conditional sampling from DiGress.

Input: Diffusion model φ_θ, predictor g_η, target y_G, guidance scale λ, graph size n
Sample $G^T \sim q_X(n) \times q_E(n)$ ▷ Random graph
for $t = T$ to 1 **do**
  $z \leftarrow f\left(G^t, t\right)$ ▷ Structural and spectral features
  $\hat{p}^X, \hat{p}^E \leftarrow \varphi_\theta\left(G^t, z\right)$ ▷ Forward pass
  $\hat{y} \leftarrow g_\eta\left(G^t\right)$ ▷ Predictor
  $p_\eta\left(\hat{y} \mid G^{t-1}\right) \propto \exp\left(\lambda \left\langle \nabla_{G^t} \log \dot{q}_\eta\left(y_G \mid G^t\right), G^{t-1} \right\rangle\right)$ ▷ Guidance distribution
  Sample $G^{t-1} \mid G^t, y_G \sim p_\theta\left(G^{t-1} \mid G^t\right) p_\eta\left(\hat{y} \mid G^{t-1}\right)$ ▷ Reverse process
end for
return G^0

The specific form of the guidance distribution is determined by the probabilistic assumptions about $\dot{q}\left(y_G \mid G^t\right)$. For example, [45] hypothesise that the target is a vector $y_G \in \mathbb{R}^d$, the distribution of which is modelled as $\dot{q}\left(y_G \mid G^t\right) = \mathcal{N}\left(g\left(G^t\right), \sigma_y I\right)$, where g is estimated by g_η, so that $\nabla_{G^t} \log \dot{q}_\eta\left(y_G \mid G^t\right) \propto -\nabla_{G^t} \|\hat{y} - y_G\|^2$, effectively leveraging the joint probability density (pdf) over all properties to enforce their simultaneous satisfaction.

4.1.1 *Conditional Sampling as Variational Inference*

While Lemma 4.1 reveals the Bayesian structure underlying the guidance mechanism by defining a posterior distribution over G^{t-1}, it does not, on its own, justify the introduction of the guidance scale λ in the exponential reweighing of the unconditional distribution during conditional sampling.

This modification can be understood as performing approximate Bayesian inference, where the exact properties' likelihood is replaced by a surrogate utility function, and the resulting distribution is obtained by solving a KL regularised variational optimisation problem. The objective takes the form:

$$\arg\max_q \left(\lambda\, \mathbb{E}_q[u(x)] - \mathrm{KL}(q(x) \| \, p(x))\right), \tag{4.3}$$

where $p(x)$ denotes the unconditional distribution and $u(x)$ is the utility function reflecting alignment with the target. The unique solution to this optimisation problem is given by:

$$q^*(x) = Z\, p(x) \exp(\lambda u(x)), \tag{4.4}$$

where Z is the normalising constant. This provides a principled justification for exponential weighting and introduces λ as a natural trade-off parameter between fidelity to the base model and preference for high-utility samples.

In our case, this framework applies directly with the following definitions:

- $x = G^{t-1}$: the graph at the previous noising step;
- $p(x) = q\left(G^{t-1} \mid G^t\right)$: the unconditional denoising distribution;
- $u(x) = \left\langle \nabla_{G^t} \log \dot{q}\left(y_G \mid G^t\right), G^{t-1} \right\rangle$: the linear approximation, up to a constant, of the log-probability that the final graph G satisfies the desired property, given the proposed sample G^{t-1}.

Substituting into Equation (4.4) gives:

$$\dot{q}\left(G^{t-1} \mid G^t, y_G\right) \approx Z\, q\left(G^{t-1} \mid G^t\right) \exp\left(\lambda \left\langle \nabla_{G^t} \log \dot{q}\left(y_G \mid G^t\right), G^{t-1} \right\rangle\right), \tag{4.5}$$

which also factorises as in Lemma 4.1 and coincides with the form adopted in our guided sampling procedure. Moreover, this distribution represents the maximum-entropy solution under the utility constraint [7], thereby mitigating over-optimisation toward the utility function, reducing mode collapse, and ultimately promoting diversity and sample quality.

4.1.2 *Monotonicity of Property Predictors under Guidance*

Having established the form of the guided reverse kernel and its dependence on λ, we now examine the implications for the evolution of the property predictors. Intuitively, increasing the guidance strength λ should bias the sampler towards candidates with higher utility values, which in our setting correspond to graphs where the properties of interest are more likely to be satisfied. The following result formalises this intuition: under mild conditions, the expected probability of each property contributing to a possibly more general rule (e.g., a cardinality rule) is a non-decreasing function of λ.

Proposition 4.1 (Monotonicity of the expectation of p_k with respect to λ)**.** *Let $y = (y_1, \ldots, y_K)$ be the properties of interest, q_λ the guided reverse kernel at guidance strength λ, and Ω_t the candidate set of graphs at time step t. Define predictors $p_j : \Omega_t \to [0,1]$, one for each property, and let $p_{rule}(p)$ denote the probability of a general rule as a function of the property predictors $p = (p_1, \ldots, p_K)$. Suppose that (i) the predictors are not negatively correlated under q_λ, i.e. $\mathrm{Cov}_{q_\lambda}(p_j, p_k) \geq 0$ for all j, k, and (ii) the rule probability p_{rule} is coordinate-wise non-decreasing in each p_j. Then, for any fixed reverse step t and any $\lambda \geq 0$, the expectation of each property predictor satisfies*

$$\frac{d}{d\lambda} \mathbb{E}_{q_\lambda}[p_k] \geq 0, \tag{4.6}$$

i.e. $\mathbb{E}_{q_\lambda}[p_k]$ is non-decreasing in λ.

The full proof is provided in Appendix B. We will later build on this result to establish further properties of our proposed guidance mechanism, introduced next to handle constraint formulations that are more flexible and expressive than individual properties or simple conjunctions.

4.2 CONDITIONAL GENERATION WITH DNF GUIDANCE

To enable interpretable and tractable integration of logical constraints into classifier-based methods—and in particular into DiGress—we propose an approach that rigorously preserves the meaning of logical formulas while remaining consistent with the probabilistic nature of diffusion models. At its core, the method treats logical satisfaction as a probabilistic event. Specifically, the satisfaction of a formula ϕ defined on a finite set of Boolean variables $\mathcal{X} = \{X_1, \ldots, X_K\}$ is modelled as a Bernoulli random variable with its parameter obtained by aggregating the Bernoulli parameters of the individual X_i's. Accordingly, the probability that the denoised graph satisfies the target formula, conditioned on the current state of the diffusion process, can be expressed as

$$\dot{q}\big(y_G \,|\, G^t\big) = \text{Bernoulli}\big(\pi\big(G^t\big)\big). \tag{4.7}$$

We approximate its expectation with a learned estimator $\pi_\eta\big(G^t\big)$, implemented via binary classifiers or regressors. The corresponding guidance gradient is then given by:

$$\nabla_{G^t} \log \dot{q}_\eta\big(y_G \,|\, G^t\big) = \big(y_G - \pi_\eta\big(G^t\big)\big) \nabla_{G^t} \operatorname{logit} \pi_\eta\big(G^t\big). \tag{4.8}$$

In general, π can be estimated if all satisfying assignments are known. To this end, we express the formula in full disjunctive normal form (DNF), which is a disjunction of conjunctions of literals corresponding to the satisfying assignments.

Definition 4.1 (Disjunctive normal form)**.** *A logical formula is in DNF if it is a disjunction of one or more disjuncts, where each disjunct is a conjunction of literals. A literal is a Boolean variable or its negation.*

Definition 4.2 (Full disjunctive normal form)**.** *A formula is in full DNF if it is the disjunction of all minterms corresponding to truth assignments that satisfy it. Each minterm is a conjunction in which every variable appears exactly once, either positively or negated.*

Remark. The full DNF representation is unique, up to the order of literals within each conjunction.

Assuming conditional independence between variables given the input (here, the graph G^t), the probability of a minterm can be approximated as the product of individual variable probabilities. Since minterms in full DNF represent disjoint assignments by definition, the overall satisfaction probability is then obtained by summing over these minterm probabilities.

4.2.1 *Cardinality Constraints in Full DNF*

In the previous section, we noted that knowing the full DNF representation of a logical formula, together with the relationship between its variables, is sufficient to compute its satisfaction probability. However, converting an arbitrary propositional formula into full DNF requires enumerating all satisfying assignments—a task that is at least as hard as solving multiple instances of the Boolean satisfiability problem (SAT). Since SAT is NP-complete [13], and enumerating all satisfying assignments may require an exponential number of cases, this process can be intractable in terms of both decision and enumeration complexity.

Therefore, instead of considering arbitrary DNF formulas, we focus on cardinality constraints—requiring at least k of K variables to evaluate to true, including pure conjunctions. Although their evaluation still reduces to an exponential number of satisfying assignments in

the worst case, their additional structure makes them considerably easier to approximate or compute in practice compared to general DNF formulas.

The following proposition establishes that cardinality constraints can always be represented in full DNF.

Proposition 4.2 (Cardinality rules in full DNF)**.** *Let $\mathcal{X} = \{X_1, \dots, X_K\}$ be a set of Boolean variables. Then, any cardinality constraint of the form*

$$\sum_{i=1}^{K} X_i \geq k \tag{4.9}$$

with $0 \leq k \leq K$, can be transformed in a logically equivalent formula in full DNF.

The proof is given in Appendix A, where the constraint is expanded into a disjunction of mutually exclusive conjunctions, each corresponding to an assignment where exactly j out of K variables are true for some $j \geq k$.

4.2.2 *Property Satisfaction Patterns under Cardinality Constraints*

In Section 4.1.2, we established that the expected value of each property predictor p_j, for all $j \in [K]$, is monotonically non-decreasing in the guidance strength λ provided that (i) the predictors are not negatively correlated and (ii) p_{rule} is non-decreasing in each p_j. The following lemma shows that condition (ii) holds for cardinality-based constraints.

Lemma 4.2 (Coordinate-wise monotonicity of cardinality rules)**.** *Fix $K \in \mathbb{N}$ and $k \in \{0, \dots, K\}$. Let $(X_1, \dots, X_K)$ be conditionally independent Bernoulli random variables with success probabilities $p = (p_1, \dots, p_K) \in [0,1]^K$. Then, the satisfaction probability of a k-of-K cardinality constraint is given by*

$$f_k(p) := \sum_{\substack{S \subseteq [K] \\ |S| \geq k}} \left(\prod_{i \in S} p_i \right) \left(\prod_{j \notin S} (1 - p_j) \right). \tag{4.10}$$

For every $j \in [K]$,

$$\frac{\partial f_k}{\partial p_j}(p) = \sum_{\substack{T \subseteq [K] \setminus \{j\} \\ |T| = k-1}} \left(\prod_{i \in T} p_i \right) \left(\prod_{\ell \notin T \cup \{j\}} (1 - p_\ell) \right) \geq 0, \tag{4.11}$$

which implies that f_k is coordinate-wise non-decreasing. Hence, whenever $f_k(p) > 0$,

$$\frac{\partial}{\partial p_j} \log f_k(p) = \frac{1}{f_k(p)} \frac{\partial f_k}{\partial p_j}(p) \geq 0. \tag{4.12}$$

Our method further establishes the monotonic tail behaviour of the expected probability that a molecule satisfies exactly r[1] conditions, and the monotonic non-decrease of the expected probability of satisfying all K conditions.

[1] We denote the number of satisfied properties by r instead of k, since this result holds for all $1 \leq r \leq K-1$, independently of value of k used to define the cardinality constraint.

Proposition 4.3 (Tail-monotonicity of expected r-of-K probabilities)**.** *Under the same assumptions as in Lemma 4.2, set $S = \sum_{i=1}^{K} X_i$ with mean $\mu_S(p) = \sum_{i=1}^{K} p_i$. For $1 \le r \le K-1$, define $p_r(p) := \mathbb{P}(S = r|p)$, the Poisson–binomial pmf at r. Furthermore, assume that $\sigma^2_{\mu_S} = \mathbb{V}_{q_\lambda}[\mu_S] < +\infty$. Then, under the same conditions as in Proposition 4.1, for every r and any $c > 0$ we obtain:*

- $\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_r] \ge 0$ *whenever* $\mathbb{E}_{q_\lambda}[\mu_S] \le r - 1 - c\sigma_{\mu_S}$, *and*
- $\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_r] \le 0$ *whenever* $\mathbb{E}_{q_\lambda}[\mu_S] \ge r + 1 + c\sigma_{\mu_S}$,

up to a tail probability of order $\frac{1}{c^2}$ by Chebyshev's inequality.

Proposition 4.4 (Monotonicity of the expected K-of-K probability)**.** *Under the guided kernel q_λ, and the same assumptions as in Lemma 4.2 and Proposition 4.1, the expectation of $p_K(p) = \prod_{j=1}^{K} p_j$, which denotes the probability that all K conditions are satisfied, is non-decreasing in λ:*

$$\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_K] \ge 0. \tag{4.13}$$

Consequently, the maximum is attained in the limit $\lambda \to \infty$.

Remark. Proposition 4.4 establishes that the expected probability of satisfying all K properties is monotonically non-decreasing, but it does not guarantee that its limit equals 1.

Tail monotonicity implies that, with high probability, the expected probability of generating graphs with exactly r satisfied conditions increases with λ at least whenever the expected number of properties is below $r - 1 - c\sigma$, and decreases with λ at least whenever it is above $r + 1 + c\sigma$. Combined with the monotonicity of the expected number of properties (which follows from the monotonicity of each expected p_k), this implies that such behaviour occurs exactly once in expectation. Moreover, the expected probability of observing graphs with all K conditions satisfied is monotonically non-decreasing with λ. These results, proved in Appendix B, characterise the satisfaction patters of any cardinality constraint p_{rule} under our method, and clearly distinguish it from the baseline guidance mechanisms introduced in the next chapter.

5 EXPERIMENTAL METHODOLOGY FOR LOGICAL GUIDANCE

This chapter presents the experimental setup and evaluation methodology used to assess the proposed logical guidance mechanism for molecular graph generation. In particular, it defines the main task, the baseline models for comparison, and the classifier architectures. It then details the dataset and preprocessing procedures, the sampling configurations for guided generation, and the metrics employed to ensure consistent and meaningful comparisons across guidance methods.

5.1 CONDITIONAL GENERATION TARGETING LIPINSKI'S RULE

The main experiment applies the logical guidance mechanism described in Section 4.2 to steer the unconditional DiGress model toward generating molecules that satisfy Lipinski's Rule of Five (hereafter, Lipinski's rule), which defines empirical criteria associated with oral bioavailability.

Definition 5.1 (Lipinski's rule)**.** *A molecule satisfies Lipinski's rule if it meets at least three of the following four criteria [27]:*

- *The partition coefficient (logP) is not greater than* 5*;*
- *The molecular weight (molWt) does not exceed* 500 *Daltons;*
- *The number of hydrogen bond donors (HBD) is not greater than* 5*;*
- *The number of hydrogen bond acceptors (HBA) is not greater than* 10*.*

From a logical perspective, we model Lipinski's rule as a cardinality constraint over four binary property indicators, satisfied whenever at least three are true.

Proposition 5.1 (Full DNF of Lipinski's rule)**.** *Let X_i denote the Boolean variable corresponding to the i-th condition. Then, Lipinski's rule has the following full DNF:*

$$\bigvee_{\substack{S\subseteq[4]\\|S|=3}} \left(\bigwedge_{i\in S} X_i \wedge \bigwedge_{j\notin S} \neg X_j \right) \vee \bigwedge_{i=1}^{4} X_i. \tag{5.1}$$

The probability that a molecule satisfies Lipinski's rule is given in the following proposition:

Proposition 5.2 (Probability of satisfying Lipinski's rule)**.** *Assume that the Bernoulli variables $(X_1, \ldots, X_4)$ are conditionally independent given the noisy graph G^t, and let $p_i = \mathbb{P}(X_i = 1 \,|\, G^t)$ be the conditional probability that the i-th condition holds on the clean graph G. Then, the probability that G satisfies Lipinski's rule given G^t is*

$$p_{Lipinski} = \mathbb{P}\left(\sum_{i=1}^{4} X_i \geq 3 \,|\, G^t \right) = \sum_{\substack{S\subseteq[4]\\|S|=3}} \left(\prod_{i\in S} p_i \prod_{j\notin S} (1-p_j) \right) + \prod_{i=1}^{4} p_i. \tag{5.2}$$

5.2 BASELINE GUIDANCE MECHANISMS

Identifying suitable baseline methods for comparison poses a methodological challenge in itself. Existing approaches are often limited to scalar conditions or lack explicit mechanisms for handling logical structure. To provide meaningful reference points for evaluation, we defined two guidance strategies that represent reasonable and practically applicable alternatives: a conjunctive rule, formulated within our probabilistic framework yet consistent with existing methodologies, and a black-box classifier guidance approach, representative of current practice. The following sections describe these two baseline strategies in detail.

5.2.1 *Conjunctive Guidance*

The conjunctive rule is a simple baseline within our probabilistic framework, defined under the conditional independence assumption, that requires all conditions to be satisfied simultaneously. Its probability is given by:

$$\dot{q}(y_G \mid G^t) = \prod_{i=1}^{K} \dot{q}(y_i \mid G^t). \tag{5.3}$$

This baseline is analogous to that tested in [45], except that their method relied on a pdf-based rather than a fully probabilistic formulation, as it operated on Gaussian variables whose exact probabilities are more difficult to estimate.

By comparing the conjunctive baseline with our logical approach, we aim to assess whether explicitly modelling the satisfaction probability of a cardinality constraint yields a distribution of generated molecules distinct from strict conjunctive compliance. In particular, we examine whether our method can enforce the rule while preserving a broader set of molecules that satisfy only k out of K properties—a diversity that may may be valuable in downstream applications where partial compliance still yields viable candidates.

5.2.2 *Black-box Rule Satisfaction Guidance*

This baseline compares our logical guidance approach with a black-box predictor that directly estimates rule satisfaction as a binary outcome, for both general cardinality constraints and the full conjunction, without relying on individual property labels. The purpose of this comparison is to assess not only whether both methods can enforce the cardinality rule effectively, but also whether they induce different distributional outcomes among satisfying molecules, thereby clarifying the trade-offs between interpretability and practical performance.

5.3 CLASSIFIERS' ARCHITECTURE

To evaluate the robustness of our method across architectural variations, and to assess their impact on the guidance mechanism, we adopt a classifier identical to the Graph Transformer used in DiGress, but reduce its depth to two layers to mitigate overfitting to our relatively simple target properties.

We experiment with three configurations: (i) independent-2L, consisting of four independently trained classifiers (one per property); (ii) shared-1L, a partially shared classifier with one shared Transformer layer followed by an independent layer and property-specific output heads;

and (iii) shared-2L, a fully shared classifier in which all Transformer layers and the output head are shared across properties. These architectures are tested for both the conjunctive baseline and our proposed logical guidance method. For the black-box guidance baseline, we use the same classifier architecture as in (i), but repurpose it to directly predict the satisfaction of Lipinski's rule and the full conjunction as binary targets, without intermediate property-level supervision.

5.3.1 *Architecture and Training Hyperparameters*

All classifiers share the same Graph Transformer backbone. Each block operates on node, edge, and global features with hidden dimensions of 256, 64, and 64, respectively, using 8 attention heads. The corresponding feedforward networks and input MLPs expand these representations to 256, 128, and 128 dimensions for nodes, edges, and global features, respectively, before projecting them back to their original sizes. No additional molecular features are included, ensuring that the architecture relies solely on the standard graph representation. Residual connections are used throughout to preserve dimensional consistency across updates.

All classifiers are trained using consistent hyperparameters (200 epochs and a batch size of 128), ensuring a uniform training setup that enables fair performance comparison.

5.4 THE GUACAMOL DATASET

Among the available pre-training datasets for DiGress (MOSES [37], GuacaMol [4], QM9 [38])[1], GuacaMol was the only one where the generated molecules had the potential to demonstrate a measurable improvement in compliance to the Lipinski's rule relative to the training data. While MOSES and QM9 consist almost exclusively of molecules satisfying all four Lipinski's conditions, GuacaMol exhibits a broader distribution of drug-like compounds, with 9.3% violating at least two criteria. Evaluation of the pre-trained models released by the authors[2] showed that molecules generated from MOSES and QM9 were consistently Lipinski-compliant, whereas 8.3% of those generated from GuacaMol were not. This setting thus allows for a more meaningful assessment of the model's capacity to shift the molecular distribution toward greater drug-likeness.

5.4.1 *Dataset Preparation and Property Statistics*

The experiments are conducted on a filtered and reconstructed version of the GuacaMol dataset. The original dataset[3] contains 1,591,378 molecules in SMILES format. To ensure compatibility with graph-based models and accurate property extraction, we retain only connected molecules that can be (i) converted to molecular graphs, (ii) reconstructed back to SMILES, and (iii) sanitised using RDKit [6]. During reconstruction, formal charges are corrected using valence-based heuristics when necessary [21], so the final dataset is based on these corrected SMILES rather than the original ones.

The resulting filtered dataset comprises 1,398,213 SMILES[4], indicating that approximately 12% of the original entries were excluded. This preprocessing ensures that Lipinski's properties

[1] Accessed at https://github.com/cvignac/DiGress/blob/main/README.md in March 2025.
[2] Accessed at https://github.com/cvignac/DiGress/tree/main/generated_samples in March 2025.
[3] Downloaded from https://figshare.com in March 2025.
[4] Split into 1,118,633 training, 69,926 validation, and 209,654 test molecules.

computed from the filtered SMILES align with those of the corresponding molecular graphs used during training. It also guarantees consistency in evaluating Lipinski's compliance on molecules generated by the model and post-processed into sanitised SMILES.

Table 1 reports descriptive statistics on the property combinations that satisfy Lipinski's rule, which will be referenced in later analyses.

Table 1: Proportions of binarised property combinations in the training set that satisfy Lipinski's rule. Patterns correspond to molWt, logP, HBD, and HBA in order. The last two rows report overall compliance with the conjunctive rule (1111) and with Lipinski's rule.

Pattern	**Proportion**
1110	0.002
1101	0.005
1011	0.109
0111	0.068
Conjunction	0.727
Lipinski	0.911

5.5 PLAN OF EXPERIMENTS

We design our evaluation so that all results are analysed as a function of the guidance strength λ, introduced in Section 4.1.1, which controls the influence of the classifier on the generative process. For both the proposed logical guidance framework and the baseline approaches, λ is varied logarithmically over the range $[0, 500]$, with six non-zero values $\{10.00,\ 21.87,\ 47.82,\ 104.56,\ 228.65,\ 500.00\}$—in addition to the unguided case ($\lambda = 0$).

This systematic variation allows us to characterise trade-offs across multiple evaluation criteria. For each λ, we record the rule compliance rate alongside the GuacaMol benchmark metrics (capturing validity, distributional similarity, and related properties) [4] and molecular diversity scores, producing a set of points in a multi-objective metric space. From this space, Pareto-optimal points are extracted—after rounding values to two decimal places to reduce stochastic variability—to identify guidance strengths that offer the best trade-offs among competing objectives. In addition, it enables us to analyse the distributional composition of Lipinski's properties at different guidance strengths, which we use to compare alternative guidance rules and classifier architectures.

The experiments span four guidance setups:

1. The unconditional DiGress model (no guidance);
2. Baseline conjunctive guidance (all four Lipinski's properties);
3. Proposed cardinality-rule guidance (at least three of four Lipinski's properties);
4. Black-box guidance, where a binary classifier predicts rule satisfaction directly.

Each active guidance setup is evaluated with three classifier architectures —independent-2L, shared-1L, and shared-2L—and, for black-box guidance, with the simplified binary predictor.

This design enables three types of comparison:

- *Impact of guidance strength:* examining how variations in λ affect the distribution of generated molecules under different constraint types and classifier architectures.
- *Role of constraint type:* comparing conjunctive and cardinality-rule guidance to assess the impact of explicitly modelling the rule's logical structure on the distribution of generated molecules.
- *Architecture effect:* comparing independent-2L, shared-1L, and shared-2L predictors under the same rule and guidance strength to assess how the classifier architecture interacts with the guidance mechanism.
- *Black-box vs. logical guidance:* comparing the simplified binary predictor against our proposed approach under the same rule to evaluate differences in the generated distributions.

For each experimental configuration and λ value, molecules are sampled until $5,000$ valid ones are obtained, as required by the GuacaMol benchmark, using 500 denoising steps. Unlike the original implementation [45], our guidance is applied on a per-molecule basis rather than per batch, enabling more precise control over the denoising process.

5.6 EVALUATION METRICS

To evaluate the quality and properties of the generated molecules, we rely on several metrics that capture constraint compliance, distributional characteristics, and chemical validity:

- *Rule satisfaction rate.* The proportion of generated molecules that satisfy either the conjunctive rule or Lipinski's rule, depending on the specific constraint enforced in each experiment. This metric serves as the primary indicator of constraint compliance.
- *GuacaMol benchmark metrics.* These include [4]:
  - *Validity.* The proportion of molecules that are parseable and sanitised by RDKit.
  - *Uniqueness.* The proportion of valid molecules that are unique within the generated set.
  - *Novelty.* The proportion of unique molecules not present in the training set.
  - *KL score.* Computed over a set of nine physicochemical descriptors, several of which correspond to the Lipinski's properties. For each descriptor, the empirical distributions of the generated and training molecules are compared using continuous or discrete KL divergence, as appropriate. An additional, non-standard KL term is computed over the distributions of internal pairwise Tanimoto similarity scores to assess structural diversity. The final score is obtained by applying an exponential transformation to each divergence value and averaging them, yielding a value in the range $[0, 1]$, where higher values indicate closer alignment to the training distribution.
  - *Fréchet ChemNet Distance (FCD) score.* Computed as the Fréchet distance between the multivariate Gaussian distributions of ChemNet feature embeddings for generated and training molecules, comparing their means and covariances [14]. An exponential transformation is applied to produce a final score in the range $[0, 1]$, where higher values indicate greater similarity.

- *Distributional shift.* To assess whether our cardinality-rule guidance induces a statistically significant and directionally meaningful shift in the distribution of Lipinski-relevant properties, we compare it against baseline methods and across classifiers trained on the same rule. The following analyses are applied:
  - *Chi-squared test on binary Lipinski's patterns.* A Pearson's Chi-squared test on a 2×16 contingency table compares the frequencies of all 4-bit Lipinski's satisfaction patterns across generated datasets, assessing whether the distributions differ significantly.
  - *Chi-squared residual analysis.* To identify which Lipinski's satisfaction profiles are most influenced by guidance, we examine the standardised residuals from the Chi-squared test for each 4-bit pattern in the guided group. Positive values indicate over-representation relative to the baseline model, and negative values indicate under-representation, highlighting the patterns most affected by the guidance mechanism.
  - *Maximum Mean Discrepancy (MMD) permutation test.* A kernel-based non-parametric test used to assess differences in the distributions of normalised Lipinski's descriptors across generated datasets. MMD measures the distance between the kernel mean embeddings of two samples in a Reproducing Kernel Hilbert Space (RKHS), capturing differences not only in means but also in higher-order moments such as variance and skewness [15]. A Gaussian kernel is used for its *universality*—the ability to distinguish any two distributions given sufficient data. The kernel bandwidth is selected using the median heuristic.
- *Molecular diversity.* To assess the structural variety of generated molecules and their similarity to the training distribution, we consider the following Tanimoto-based metrics [1]:
  - *Internal diversity.* Defined as one minus the average pairwise Tanimoto similarity between generated molecules, computed using Morgan fingerprints (radius 2, 4096 bits, as used in the GuacaMol benchmark). This metric quantifies structural diversity within the generated set: higher values indicate more variation, while lower values suggest redundancy or mode collapse.
  - *External diversity.* Defined as one minus the average Tanimoto similarity between each generated molecule and its most similar counterpart in the training set, computed using the same Morgan fingerprints as in internal diversity. This metric captures how novel the generated molecules are with respect to the training distribution—higher values indicate greater deviation from the training set, while lower values suggest closer alignment.
- *Latent Space Analysis.* For visualisation, we subsample $1,000$ molecules to maintain plot clarity.
  - *Uniform Manifold Approximation and Projection (UMAP).* We visualise the distribution of generated molecules using UMAP [18] applied to Continuous Data-Driven Descriptors (CDDD) [50]. These pre-trained embeddings capture structural and physicochemical features in a data-driven way and generalise well due to training on large molecular datasets. To ensure consistent layout across comparisons, the UMAP is fit on training set embeddings and used to project molecules from different guidance strategies. We use 15 neighbours to balance preservation of local and global

structure, a minimum distance of 0.1 to allow moderate clustering in the 2D projection, and cosine distance, which is well-suited for high-dimensional embeddings like CDDD due to its focus on angular rather than magnitude-based similarity. Points are coloured based on compliance with the target rule, making it easier to see the distribution of chemically plausible molecules.

– *Principal Component Analysis (PCA).* We apply PCA to standardised Lipinski's properties to visualise how their distributions shift under guidance. PCA projects the data onto orthogonal axes that capture the most variance, producing a 2D plot where shifts in molecular profiles are interpretable. As a linear method, PCA provides meaningful loadings that reveal how each property contributes to the principal components. The model is fit on the training set or on DiGress outputs and then applied to the overlaid samples. Points are coloured by compliance with the target rule, aiding the interpretation of chemical plausibility.

Taken together, these metrics provide a comprehensive toolkit for evaluating guided molecular generation. While not all metrics are reported simultaneously (in practice we select the most informative subset for each comparison), they collectively offer a multifaceted perspective on model performance, capturing constraint satisfaction, distributional alignment, chemical validity, and structural diversity.

6 RESULTS AND DISCUSSION: A LIPINSKI'S CASE STUDY

In this chapter, we present and discuss the experimental results. We first use the unconditional DiGress model as a reference, then analyse the impact of guidance strength across different rules, compare conjunctive and Lipinski's rule guidance, assess the influence of classifier architectures, and conclude with a black-box guidance experiment supporting the motivation for our framework. Representative uncurated samples generated under Lipinski's guidance are shown in Appendix C.3.

6.1 THE UNCONDITIONAL DIGRESS MODEL

We examine the GuacaMol benchmark and diversity metrics for the unconditional DiGress model trained on the filtered GuacaMol dataset described in Section 5.4. In addition, we compare the generated and training distributions through statistical tests and visual analyses based on UMAP and PCA.

6.1.1 *Benchmark Performance, Diversity, and Compliance*

Table 2: Distribution-learning benchmark metrics for DiGress.

Metric	**Score**
Validity	0.933
Uniqueness	0.999
Novelty	1.000
KL	0.948
FCD	0.583

Table 3: Tanimoto-based diversity metrics for DiGress.

Metric	**Score**
Internal Diversity	0.894
External Diversity	0.682

Tables 2 and 3 report standard GuacaMol benchmarks and additional molecular diversity scores. Notably, our results generally outperform those originally reported by the authors on this dataset [45], likely due to corrections later integrated into the official code repository, as the version used in the paper is unspecified. We achieve higher validity (0.933 vs. 0.852) and KL scores (0.948 vs. 0.929), while obtaining a lower FCD score (0.583 vs. 0.680), suggesting closer property distributions but residual shifts in molecular embeddings. The diversity metrics, which were not originally reported, further show that the model does not suffer from mode

collapse, as indicated by strong internal Tanimoto diversity, while retaining only moderate similarity with the training set, as reflected by external diversity.

6.1.2 *Distributional Shift*

Table 4: Proportions of binarised property combinations in the DiGress samples that satisfy Lipinski's rule. Patterns correspond to molWt, logP, HBD, and HBA in order. The last two rows report overall compliance with the conjunctive rule (1111) and with Lipinski's rule.

Pattern	**Proportion**
1110	0.003
1101	0.012
1011	0.099
0111	0.064
Conjunction	0.749
Lipinski	0.927

Compared to the original dataset, DiGress induces slight shifts in the raw Lipinski's statistics, significantly altering the training distribution according to the MMD ($p \approx 10^{-7}$) and Chi-squared ($p \approx 10^{-8}$) tests. Table 4 reports the proportions of binarised property patterns. The most frequent pattern is 1111, which increases compared to the training set (0.749 vs. 0.727). The next most common patterns, 1011 (0.099 vs. 0.109) and 0111 (0.064 vs. 0.068), occur slightly less often in DiGress. In contrast, pattern 1101 is more frequent in DiGress (0.012 vs. 0.005). Consequently, Lipinski's rule compliance improves from 0.908 to 0.927.

6.1.3 *Latent Space Analysis*

To characterise the distribution of generated molecules, we visualise UMAP and PCA embeddings for the DiGress model. In Figure 7, UMAP shows that the unconditional DiGress model broadly covers the training space, preserving global and local structure without evidence of over-concentration or mode collapse.

The PCA visualisation in Figure 8 shows that the property distribution of the generated molecules closely mirrors that of the training set. In particular, the first principal component (PC1, explaining 47% of the variance) is positively correlated with molWt (0.60), HBD (0.50), and HBA (0.62), capturing variation related to molecular size and hydrogen-bonding potential (polarity). The second principal component (PC2, explaining 34% of the variance) is dominated by logP (0.83), a measure of hydrophobicity, with additional positive loading from molWt (0.43) and a negative association with HBD (-0.35). Movement in the positive direction along both PC1 and PC2 corresponds to larger, more hydrophobic molecules—compounds that are less likely to satisfy Lipinski's rule.

In summary, the unconditional DiGress model provides a strong baseline, generating chemically valid and diverse molecules. While it induces slight shifts in property distributions that modestly increase overall compliance with Lipinski's rule, latent space analyses confirm its close alignment with the training distribution.

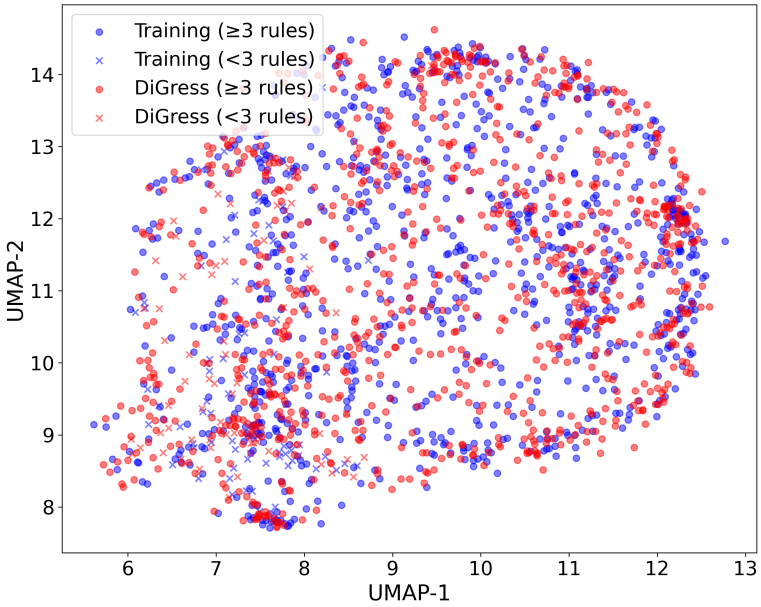


Figure 7: UMAP projection of molecules from the training set, overlaid with those generated using DiGress. All samples are projected onto the UMAP embedding learned from the training data for consistency.

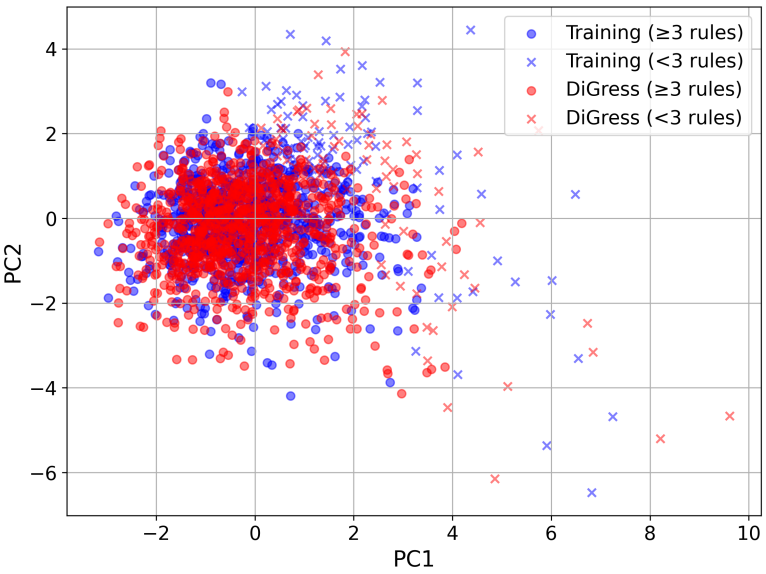


Figure 8: PCA projection of molecules from the training set, overlaid with those generated using DiGress. Both sets are projected onto the principal components learned from the training samples for consistency.

6.2 THE IMPACT OF GUIDANCE STRENGTH ON GUIDANCE RULES

The guidance strength λ controls the influence of the classifier on the generative process and is the key variable in our evaluation. Here, we analyse how varying λ affects generation quality for both conjunctive guidance and Lipinski's rule guidance, using four independently trained classifiers for clarity (configuration (i) in Section 5.3). This allows us to establish the main performance trends for each rule before introducing the additional variation due to classifier architecture.

As shown in Figure 9, increasing λ consistently improves target property satisfaction, approaching perfect compliance regardless of the rule. This improvement, however, comes at the cost of lower KL, FCD, and validity scores, highlighting the trade-off between constraint satisfaction and alignment with the training distribution. Because KL and FCD are positively correlated, their declines generally occur in tandem. The accompanying drop in validity under stronger guidance likely reflects the model's limited ability to generalise to new regions of chemical space, which contributes to overfitting to the rule. Among the two rules, conjunctive guidance exhibits the steepest decline across all values of λ, whereas Lipinski's rule degrades less sharply, preserving higher validity and closer alignment with the training distribution. This difference likely reflects the greater flexibility of Lipinski's 3-of-4 formulation, which is inherently easier to enforce than a strict conjunction and aligns more closely with the training distribution, where many molecules already satisfy three of the four properties (recall of 0.908 vs. 0.727 for those satisfying all four; see Table 1).

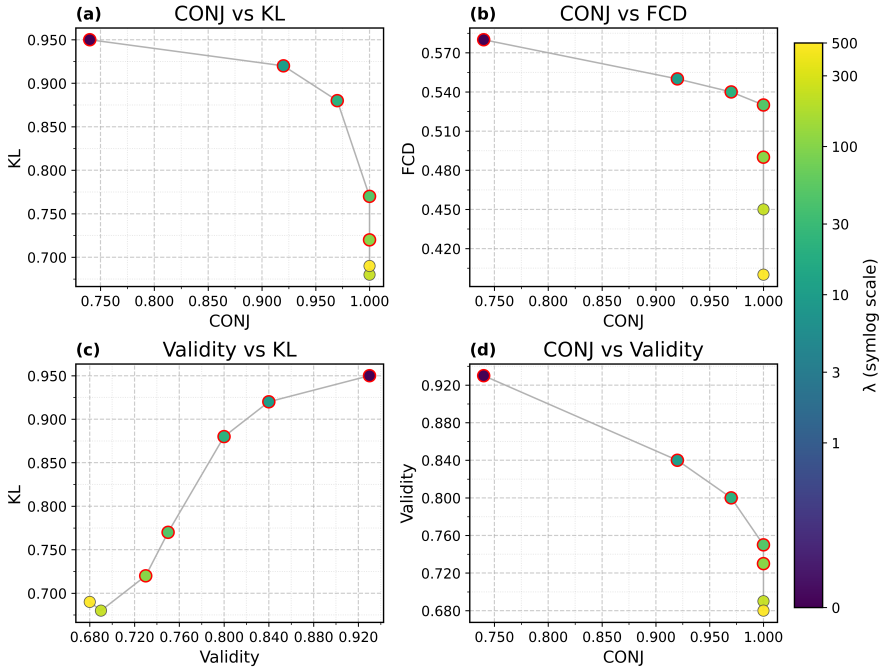
In terms of diversity, external Tanimoto diversity (relative to the training set) tends to increase slightly for both rules as λ grows—stabilising just below 0.70—mirroring the KL and FCD trends that indicate increasing divergence from the training distribution. Internal Tanimoto diversity (within the generated set) also increases, reaching up to 0.90, indicating no evidence of mode collapse across the tested λ values.

The Pareto frontier analysis (Figure 9, red-circled points) shows that low to intermediate values of λ in the range 10–104 achieve the most favourable trade-offs, combining high KL and FCD scores with strong validity and near-perfect compliance. These settings also maintain high novelty, uniqueness, and Tanimoto diversity—metrics that are included in the Pareto frontier computation but are not shown in the figure—remaining close to the DiGress reference. Notably, the unconditional DiGress model lies on the Pareto frontier for both guidance rules, offering a distinct trade-off that maximises validity and distributional similarity without applying explicit constraints.

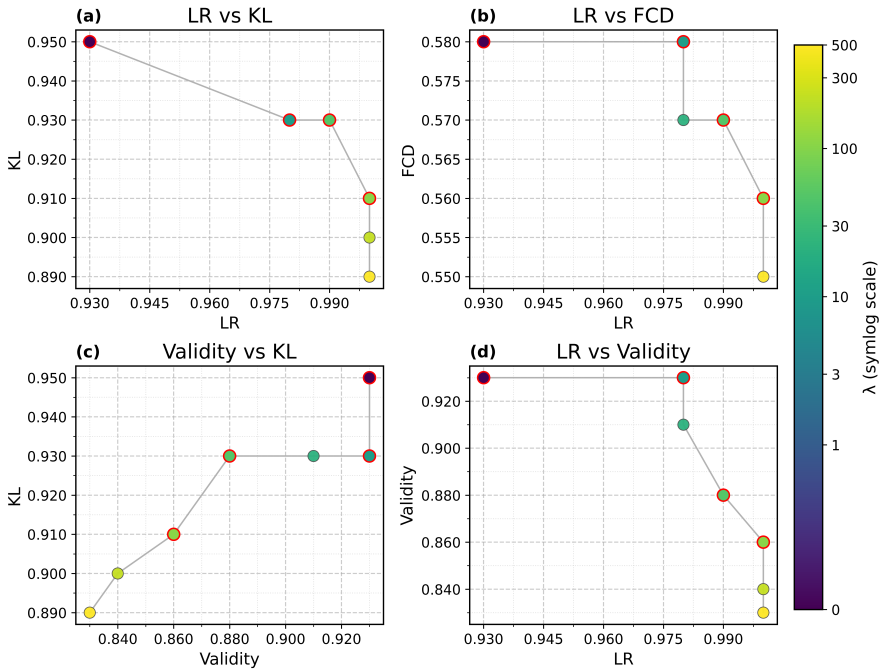
In summary, increasing λ improves compliance without reducing diversity, but decreases validity and distributional similarity, with conjunctive guidance degrading more sharply than Lipinski's rule. Low to intermediate λ values yield the most favourable trade-offs, while the unconditional DiGress model remains Pareto-optimal for validity and fidelity.

6.3 CONJUNCTIVE VS. LIPINSKI'S RULE GUIDANCE

We compare conjunctive and Lipinski's rule guidance by analysing their impact on Lipinski's property distributions, with molecular embeddings visualised through UMAP and PCA to aid interpretability. For consistency, we employ four independently trained property predictors,



(a) Conjunctive guidance.



(b) Lipinski's rule guidance.

Figure 9: Trade-offs between compliance, distributional similarity (KL, FCD), and validity across guidance strengths λ (colour-coded, log scale) for the independent-2L classifiers. Red-circled points indicate the Pareto frontier.

as in the previous section, and defer the analysis of architecture-dependent effects to the next section.

6.3.1 *Distributional Shift*

To assess distributional differences, we applied statistical tests to Lipinski's property distributions. The MMD test shows significant differences across all λ values, with permutation p-values between 10^{-10} and 10^{-37} that increase with λ. Likewise, the Chi-squared test on binary property-satisfaction patterns yields extremely small p-values (10^{-147}–10^{-216}), confirming substantial differences in the frequency of property combinations between the two guidance rules.

Chi-squared residuals show that conjunctive guidance produces almost exclusively fully compliant molecules, increasing from 92.3% to 100% as λ grows, while almost eliminating partially satisfied cases—especially 3-of-4 configurations, whose frequency drops from 3.3% to 0%. In contrast, guidance by Lipinski's rule enriches intermediate patterns despite the continued dominance of full compliance, particularly at moderate values of λ (10–104), where 19.9%–18.4% of molecules fall into 3-of-4 configurations and the fully compliant share decreases to 77.9%–81.5%. This behaviour reflects a key property of our method: at moderate guidance strengths, it enhances or maintains the coverage of the minimal satisfying set of Lipinski's rule (the 3-of-4 configurations), enabled by direct gradient control.

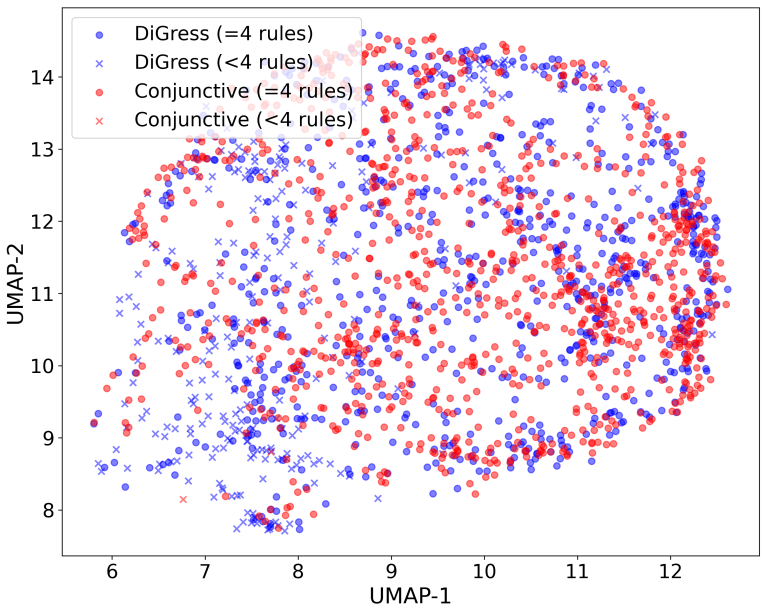
6.3.2 *Latent Space Analysis*

We visualise the distributions of molecules generated by each guidance rule with UMAP and PCA projections (Figures 10 and 11), using a guidance strength of $\lambda = 47.82$ selected from the Pareto frontier for its strong generation quality across both rules.

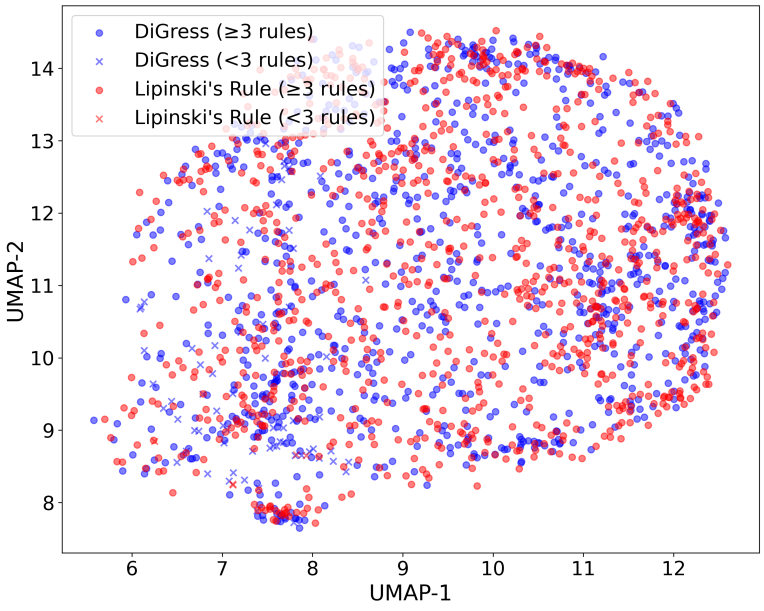
The UMAP plots reveal regional density shifts caused by property constraints, with both methods concentrating generation within narrower areas of chemical space. In the conjunctive guidance plot, dense clusters of red points appear in the mid-right region, while gaps are visible to the left, where DiGress originally generates molecules that do not comply with the rule. In contrast, samples from Lipinski's guidance remain closer to the unconditional DiGress distribution, exhibiting more uniform sampling and better coverage.

In the PCA projections, Lipinski's guidance preserves broader property variation, yielding a point cloud that is more diffuse than under conjunctive guidance and spans nearly as wide a region as the unconditional DiGress samples. Notably, the principal components fitted on the DiGress data closely resemble those previously derived from the training set in Section 6.1.3. PC1 (explaining 49% of the variance) is positively associated with molWt (0.52), HBD (0.55), and HBA (0.62), while PC2 (accounting for 34% of the variance) is driven primarily by logP (0.79), with additional loading from molWt (0.56) and an opposing contribution from HBD (-0.25).

In summary, conjunctive guidance collapses almost entirely to full compliance, eliminating 3-of-4 cases, whereas Lipinski's guidance preserves these intermediate patterns, especially at moderate values of λ. Latent space analyses confirm that enforcing Lipinski's rule yields broader coverage and closer alignment with the training distribution.

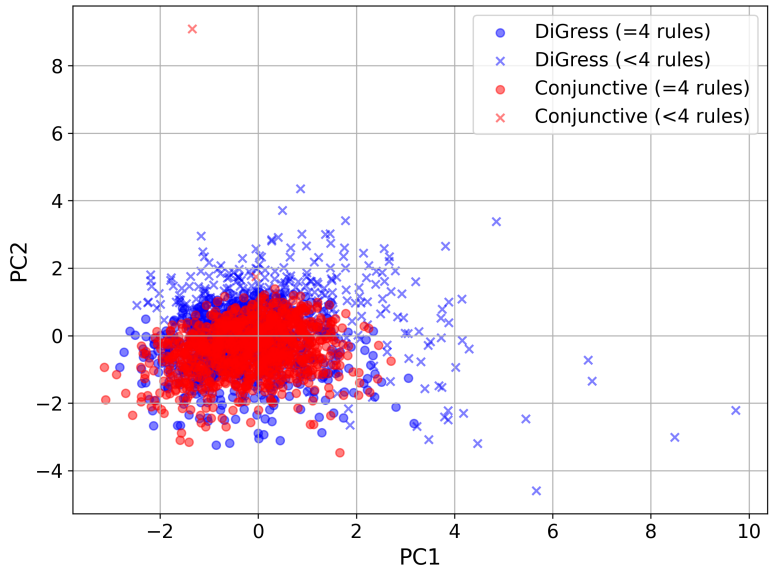


(a) Conjunctive guidance.

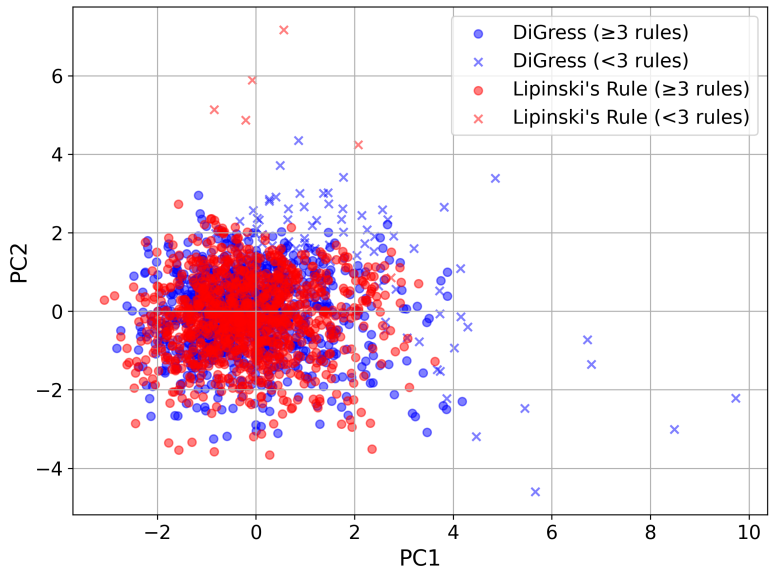


(b) Lipinski's rule guidance.

Figure 10: UMAP projection of molecules from the DiGress model, overlaid with those generated using the independent-2L classifiers. All samples are projected onto the UMAP embedding learned from the training data for consistency.



(a) Conjunctive guidance.



(b) Lipinski's rule guidance.

Figure 11: PCA projection of molecules from the DiGress model, overlaid with those generated using the independent-2L classifiers. Both sets are projected onto the principal components learned from DiGress samples for consistency.

6.4 COMPARISON BETWEEN CLASSIFIER ARCHITECTURES

We now present the comparative analysis of the three classifier architectures described in Section 5.3 when applied to logical guidance. The two extremes of layer sharing are represented by independent-2L, which has no shared Transformer layers, and shared-2L, which shares both layers across properties, with shared-1L lying in between.

To ensure a fair comparison, all classifiers were trained under the same standardised conditions (Section 5.3.1). They achieved broadly similar performance: their Precision–Recall curves and Average Precision were nearly identical across properties—reflecting comparable separation across class probabilities—with accuracy, F_1, and Area Under the ROC Curve likewise consistent. This confirms that any differences observed in the guidance mechanism are not due to disparities in classifier capacity, but rather to how the architecture couples with the model when guidance is applied.

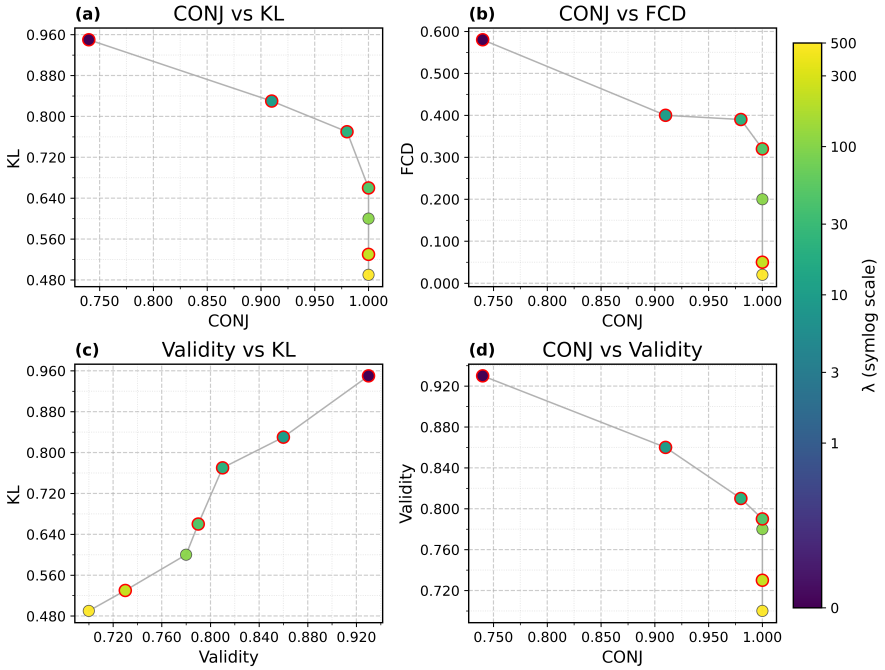
6.4.1 *Benchmark Performance, Diversity, and Compliance*

We begin by comparing independent-2L and shared-2L (Figures 9 and 12), focusing on Lipinski's rule since the conjunctive rule yields broadly similar trends. Across all λ values, shared-2L achieves consistently higher compliance and validity, but diverges on other metrics: it shows steeper drops in KL and FCD at matched compliance or validity levels, indicating greater deviation from the training distribution. This shift is accompanied by higher external Tanimoto diversity, particularly under conjunctive guidance. A likely explanation is that jointly learning the properties in the multi-head model strengthens the guidance signal, enabling broader exploration of property combinations than independent-2L, while remaining within the training support. Both architectures avoid mode collapse and maintain internal Tanimoto diversity comparable to DiGress.

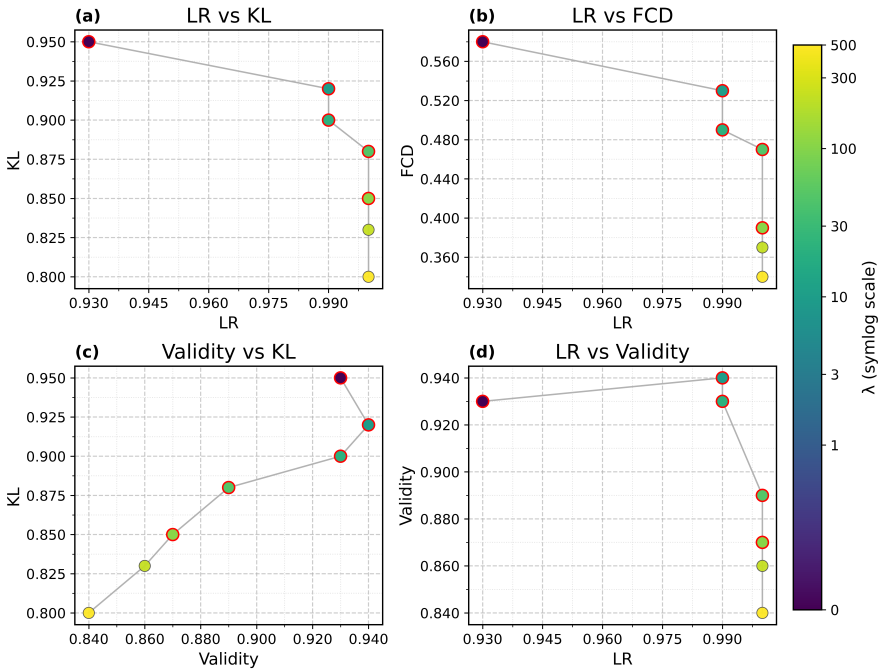
As expected, the shared-1L classifier performs intermediately between independent-2L and shared-2L in terms of rule satisfaction, validity, KL and FCD degradation, and diversity (Figure 13). When pooling results across all classifiers and guidance strengths, the Pareto frontier analysis shows that all architectures contribute efficient points, with λ values concentrated in the range 10–104. Shared-1L appears most frequently on the frontier (4 points), while the other architectures contribute two each, suggesting that shared-1L provides robust trade-offs across a wider range of guidance strengths.

6.4.2 *Distributional Shift*

We now analyse how property satisfaction patterns evolve under guidance, comparing conjunctive and Lipinski's rules as well as different classifier architectures. For all classifiers, Lipinski's rule produces a more balanced mixture of 3-of-4 and 4-of-4 configurations, whereas the conjunctive rule favours predominantly 4-of-4, more homogeneous outputs (c.f. Section 6.3.1). Notably, at intermediate values of λ (10–104), 3-of-4 molecules persist more under independent-2L (19.9%–18.4%) than under shared-1L (20%–15.6%) or shared-2L (21%–16.4%). As λ increases, the properties are increasingly driven toward joint satisfaction, amplifying the 4-of-4 mode while diminishing the prevalence of 3-of-4 cases, which fall to 17.1%, 9.9% and 11.9% respectively for the three architectures at $\lambda = 500$. This trend is reinforced both by the stronger drive toward

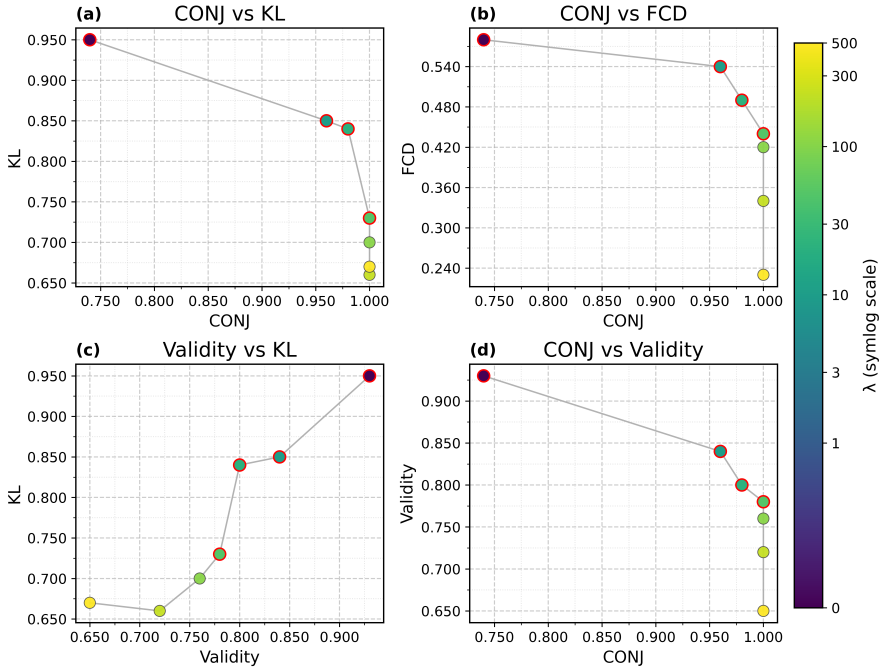


(a) Conjunctive guidance.

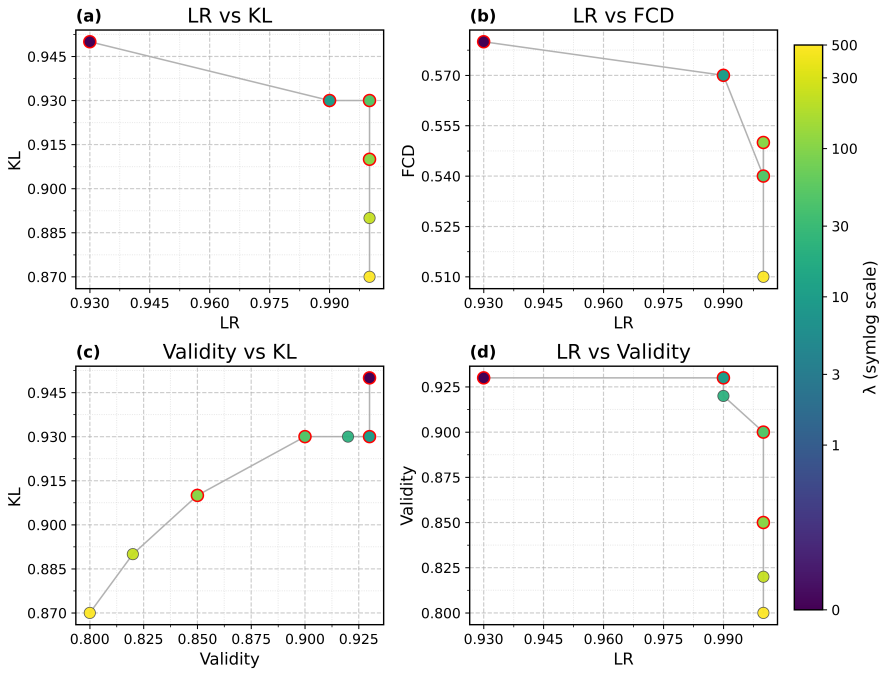


(b) Lipinski's rule guidance.

Figure 12: Trade-offs between compliance, distributional similarity (KL, FCD), and validity across guidance strengths λ (colour-coded, log scale) for the shared-2L classifier. Red-circled points indicate the Pareto frontier.



(a) Conjunctional guidance.



(b) Lipinski's rule guidance.

Figure 13: Trade-offs between compliance, distributional similarity (KL, FCD), and validity across guidance strengths λ (colour-coded, log scale) for the shared-1L classifier. Red-circled points indicate the Pareto frontier.

full satisfaction in the shared architectures and by the larger proportion of fully compliant molecules in the training set.

These empirical patterns—both the growing prevalence of the 4-of-4 mode and the persistence of 3-of-4 cases—are consistent with the theoretical properties established in Section 4.2.2. The increase of the 4-of-4 mode with λ is expected, since stronger guidance raises the probability of each property and thus the expected number of satisfied properties. By contrast, once this number exceeds three, our proposition no longer determines whether the 3-of-4 proportion should increase or decrease. A plausible explanation for the observed decrease is that the property predictors start with similar values and become more aligned as λ increases, causing the distribution to approximate a binomial. In that case, the turning point for the 3-of-4 mass coincides with the mean, which in the training dataset is already above 3.

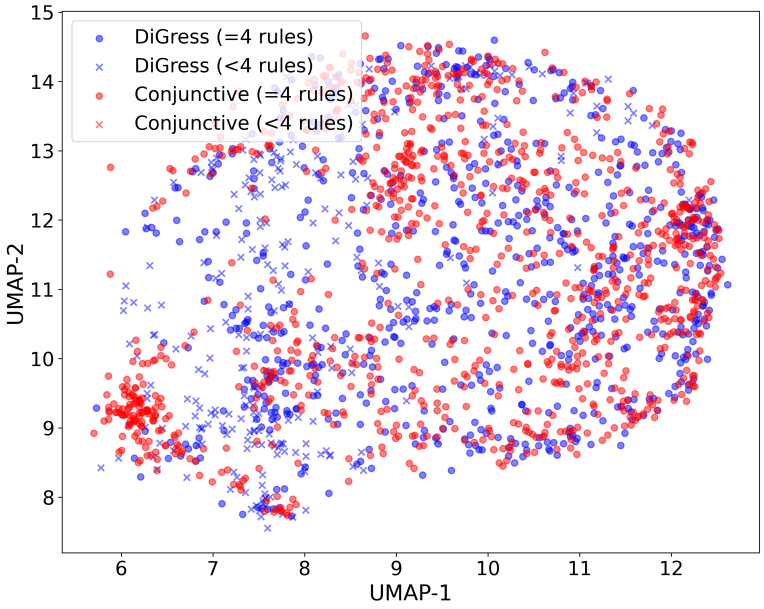
Across architectures and λ values, the most frequent 3-of-4 configuration is the combination in which molWt, HBD, and HBA are satisfied while logP remains unsatisfied (1011), accounting for 7%–13.1% of all samples. By contrast, configurations where HBA or HBD are unsatisfied while molWt is satisfied (1110 or 1101) remain rare during optimisation, each occurring in less than 1% of samples. In both cases, these patterns mirror imbalances already present in the training data and inherited by the unconditional DiGress model. Other relatively frequent configurations—such as the one with only HBD and HBA satisfied (0011), which occurs in both the training set and DiGress (4.6% and 7%, respectively)—are strongly reduced under guidance, since they do not satisfy Lipinski's rule (averaging just 0.6% across classifier architectures and λ values). The consistency of these findings across classifiers indicates that the guidance mechanism leaves trends already present in the data and unconditional model largely unaltered when they align with the optimised rule, and suppresses them when explicitly constrained by it. Importantly, the mechanism does not explicitly favour one 3-of-4 configuration over another.

6.4.3 *Latent Space Analysis*

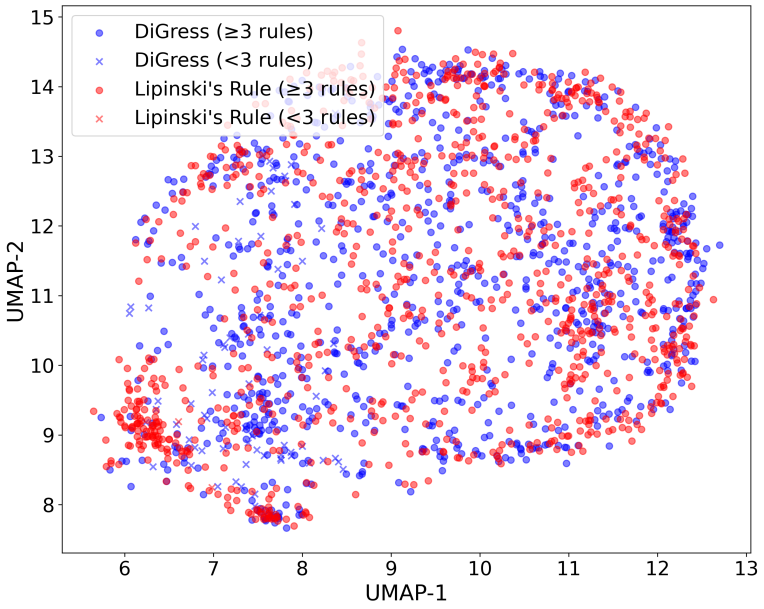
Finally, to characterise the impact of different classifier architectures, we examine UMAP projections of the generated molecules at $\lambda = 47.82$ (Figures 14 and 15). Under Lipinski's rule, shared-1L behaves similarly to independent-2L (Figure 10b), whereas shared-2L yields denser clusters of molecules—particularly in the bottom-left region and more uniformly across the embedding space—consistent with the stronger coupling of property signals in this fully shared architecture. For the conjunctive rule, shared-2L shows reduced coverage of the DiGress unconditional space, with larger uncovered areas (e.g., in the centre-left) and denser regions elsewhere (notably in the bottom-left), while shared-1L again resembles independent-2L (Figure 10a).

These projections indicate that, although overall trends across rules and architectures remain consistent, the classifiers cover distinct regions of the molecular space, suggesting that they learn slightly different distributions. This effect is harder to appreciate in PCA projections, which we include in Appendix C.1 for completeness.

Across all analyses, we find that classifier architecture affects the strength and distributional impact of guidance, but not the qualitative trends. Shared-2L provides the strongest compliance and validity at the cost of larger distributional shifts, independent-2L maintains weaker coupling between properties, and shared-1L offers intermediate behaviour with robust trade-offs. Distributional analysis shows that 4-of-4 configurations increase monotonically with λ, while 3-of-4 account for a substantial share at moderate λ but decrease because the mean number of

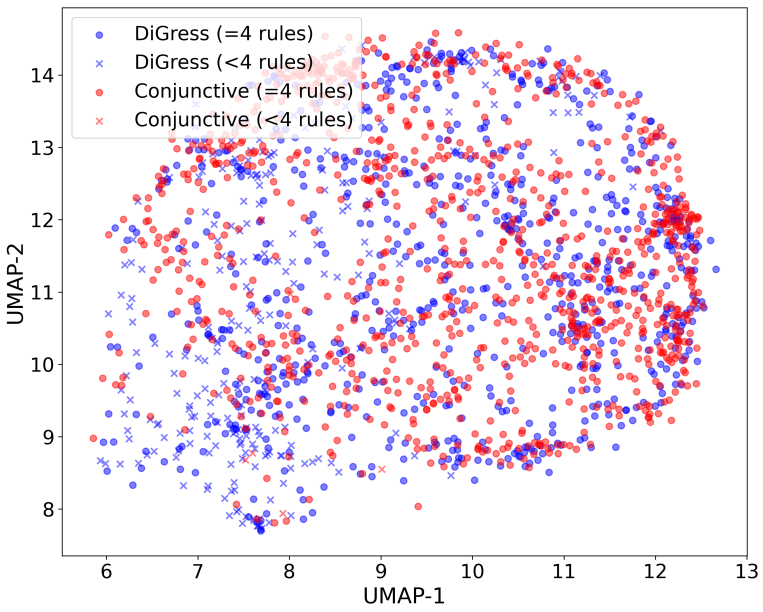


(a) Conjunctive guidance.

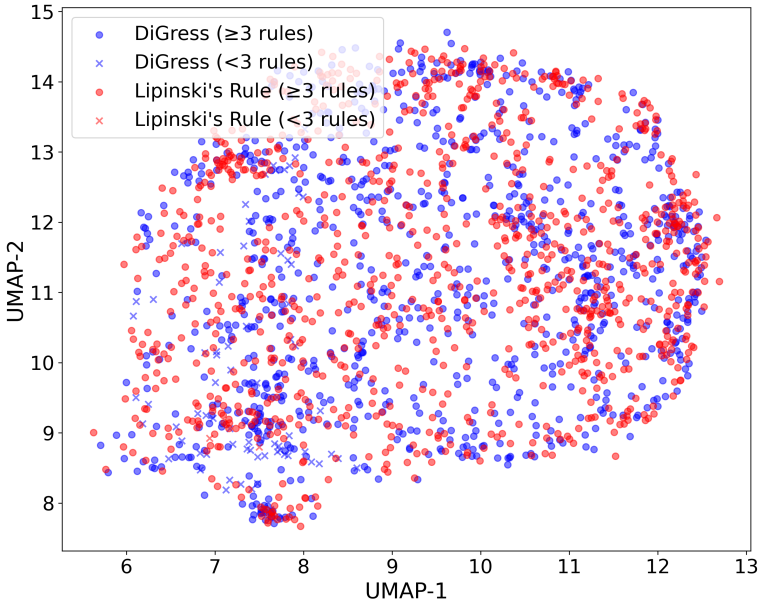


(b) Lipinski's rule guidance.

Figure 14: UMAP projection of molecules from the DiGress model, overlaid with those generated using the shared-2L classifier. All samples are projected onto the UMAP embedding learned from the training data for consistency.



(a) Conjunctive guidance.



(b) Lipinski's rule guidance.

Figure 15: UMAP projection of molecules from the DiGress model, overlaid with those generated using the shared-1L classifier. All samples are projected onto the UMAP embedding learned from the training data for consistency.

satisfied properties exceeds 3. Latent space projections confirm that each architecture explores distinct regions of molecular space, while broadly preserving patterns present in the training distribution and eliminating those incompatible with the rule.

6.5 BLACK-BOX VS. LIPINSKI'S RULE GUIDANCE

In this section, we compare our logical guidance method with a black-box classifier that predicts binary rule compliance directly, as introduced in Section 5.2.2. We summarise the main findings here and defer the complete set of results to Appendix C.2.

This model achieves perfect compliance under both the conjunctive and Lipinski's rules, with trade-offs in validity, KL, FCD, and diversity that remain broadly comparable to those of logical guidance as λ varies. However, under the conjunctive rule, it exhibits a less favourable KL–FCD vs. validity trade-off and reduced novelty (down to 95.5% at $\lambda = 500$, compared to an average of 99.9% for logical guidance).

Although the black-box baseline may appear to preserve a higher proportion of 3-of-4 configurations under Lipinski's rule—rising from 19.4% to 26.1% as λ increases—a closer inspection reveals shortcut learning. When extending λ up to 1500, the only configuration that consistently grows (reaching 31.4%) is 1011, while 1111, which also satisfies the rule, steadily declines to 63.9%. This indicates that the model focuses on satisfying the three properties most frequently observed in the dataset—molWt, HBD, and HBA—which are exactly the ones shared by the satisfying configurations 1011 and 1111. As a result, it fails to capture the intended meaning of the cardinality rule "*at least* three of four", effectively optimising exactly three properties and neglecting part of the satisfying region. Therefore, we can conclude that, in general, a principled enforcement of the logical rule—where guidance is applied explicitly to each property, as in our method—is necessary to achieve the desired behaviour.

In summary, while both black-box and logical guidance achieve high compliance, they differ in how satisfying configurations are distributed. For Lipinski's rule, the black-box baseline falls into shortcut learning, overfitting to the most common set of three properties underlying the patterns that satisfy the rule, whereas our method preserves the probabilistic and logical meaning of the original rule.

7 CONCLUSIONS

This research set out to address the absence of principled methods for enforcing logical constraints in conditional diffusion models for graphs. To this end, it pursued five main objectives: to identify the research gap and select a suitable base model; to define a representative conditional generation task and benchmark; to develop a probabilistic framework for logical guidance; to implement and evaluate it experimentally; and to analyse its performance and limitations.

Building on DiGress, we introduced a framework that redefines conditional graph generation by embedding logical rules, expressed in full disjunctive normal form, into diffusion models through a probabilistic treatment of their satisfaction. The resulting guidance mechanism, grounded in minimal assumptions, enables tractable and interpretable sampling under logical filters. Focusing on cardinality constraints—such as Lipinski's Rule of Five, a cornerstone of drug discovery—and evaluating them on the GuacaMol benchmark, we demonstrated that the framework effectively enforces domain-relevant rules while preserving their intended semantics. To our knowledge, this constitutes the first rigorous formulation of logical guidance at scale, establishing a methodological precedent for conditional molecular generation.

Overall, the work provides a principled foundation for integrating logical reasoning into generative diffusion models, while several assumptions and limitations naturally suggest directions for further development.

7.1 FUTURE WORK

Looking ahead, several methodological directions could broaden this framework. Extending logical guidance beyond DiGress—for example to score-based SDEs—would require new theoretical tools, as the discrete structure underpinning our proofs no longer applies. Another challenge is to trace the origins of constraint overfitting—whether in the diffusion model, the guidance mechanism, or their interaction—and to design strategies that mitigate it while sustaining a rich space of valid samples. Finally, probing the limits of the conditional independence assumption used to compute rule probabilities—and ultimately relaxing it—would bring the framework closer to reality and reduce systematic bias. Alongside these theoretical challenges, experimental refinements can further strengthen the framework's applicability.

On the practical side, finer control on the generated samples could be achieved by modelling the satisfaction pattern as a latent variable, combining a rigorous probabilistic foundation with explicit control over the prevalence of samples that satisfy exactly k or all K conditions, as well as over which configurations arise within the k-of-K case. Another practical challenge is scalability: the effectiveness of the proposed probabilistic implementation of cardinality constraints in high-dimensional settings remains to be tested empirically. As the number of properties increases, correlations among variables are likely to become more complex, making the outcomes harder to interpret, both experimentally and theoretically. Consequently, new strategies may need to be developed to handle cases involving many properties.

Overall, this thesis showed that logical rules can be embedded directly into the generative process, laying the groundwork for probabilistic–symbolic integration in graph diffusion models. This points toward a new generation of models that combine power with transparency and control, with impact extending far beyond molecular design.

A

DNF REPRESENTATION OF CARDINALITY CONSTRAINTS

In this chapter we show that any cardinality constraint of the form $\sum_{i=1}^{K} X_i \geq k$ can be rewritten as a logically equivalent formula in full DNF. The following proposition formalises this result.

Proposition A.1 (Cardinality rules in full DNF)**.** *Let $\mathcal{X} = \{X_1, \ldots, X_K\}$ be a set of Boolean variables. Then, any cardinality constraint of the form*

$$\sum_{i=1}^{K} X_i \geq k \tag{A.1}$$

with $0 \leq k \leq K$, can be transformed in a logically equivalent formula in full DNF.

Proof. The cardinality constraint is satisfied if and only if at least k of the variables in $\mathcal{X}$ are realised, which we denote as the event E. In probability theory, this condition is equivalent to the union of the assignments where exactly j of the K variables are true and the remaining $K - j$ are false, which we denote as S_j. The number of such assignments is $\binom{K}{j}$.

$$E := \left(\sum_{i=1}^{K} X_i \geq k\right) \equiv \bigcup_{j=k}^{K} S_j := \bigcup_{j=k}^{K} \left(\bigcup_{\substack{A \subseteq \mathcal{X} \\ |A|=j}} \left(\bigcap_{X_i \in A} X_i \cap \bigcap_{X_i \in \mathcal{X} \setminus A} \neg X_i \right) \right). \tag{A.2}$$

Therefore, the event E can be expressed as a union of intersections of elementary events. Furthermore, since each conjunction:

- contains all K variables, each appearing exactly once (either positively or negated), and
- corresponds to a unique truth assignment, and no two conjunctions overlap (i.e., they represent disjoint assignments),

the resulting formula is precisely the full DNF of the cardinality constraint. □

B

THEORETICAL ANALYSIS OF SATISFACTION PATTERNS WITH CARDINALITY CONSTRAINTS

In this chapter we present a theoretical analysis of property satisfaction patterns with cardinality rule guidance. We first show analytically that the satisfaction probability of cardinality rules is non-decreasing in each predictor, and that, under mild assumptions, applying guidance ensures the expected predictor for each property is also non-decreasing in λ. Then, we prove that the tail behaviour of the expected probabilities of graphs satisfying exactly r properties is monotonic in λ, and that the expected probability of satisfying all K properties is monotonically non-decreasing.

Lemma B.1 (Coordinate-wise monotonicity of cardinality rules)**.** *Fix $K \in \mathbb{N}$ and $k \in \{0, \dots, K\}$. Let $(X_1, \dots, X_K)$ be conditionally independent Bernoulli random variables with success probabilities $p = (p_1, \dots, p_K) \in [0,1]^K$. Then, the satisfaction probability of a k-of-K cardinality constraint is given by*

$$f_k(p) := \sum_{\substack{S \subseteq [K] \\ |S| \geq k}} \left(\prod_{i \in S} p_i \right) \left(\prod_{j \notin S} (1 - p_j) \right). \tag{B.1}$$

For every $j \in [K]$,

$$\frac{\partial f_k}{\partial p_j}(p) = \sum_{\substack{T \subseteq [K] \setminus \{j\} \\ |T| = k-1}} \left(\prod_{i \in T} p_i \right) \left(\prod_{\ell \notin T \cup \{j\}} (1 - p_\ell) \right) \geq 0, \tag{B.2}$$

which implies that f_k is coordinate-wise non-decreasing. Hence, whenever $f_k(p) > 0$,

$$\frac{\partial}{\partial p_j} \log f_k(p) = \frac{1}{f_k(p)} \frac{\partial f_k}{\partial p_j}(p) \geq 0. \tag{B.3}$$

Proof. To compute the partial derivative of f_k with respect to p_j, it is convenient to separate the subsets $S \subseteq [K]$ into those containing j and those that do not. For each $T \subseteq [K] \setminus \{j\}$, consider the pair

$$S_1 = T \cup \{j\}, \qquad S_0 = T. \tag{B.4}$$

The corresponding terms share the same product over the other coordinates, with S_1 contributing a factor p_j and S_0 a factor $(1 - p_j)$. Differentiating therefore yields a $+1$ contribution from S_1 if $|S_1| \geq k$, and a -1 contribution from S_0 if $|S_0| \geq k$. Thus, the combined contribution of the pair (S_0, S_1) is

$$\mathbf{1}\{|T| \geq k-1\} - \mathbf{1}\{|T| \geq k\} = \mathbf{1}\{|T| = k-1\}, \tag{B.5}$$

so only subsets T of size $k-1$ contribute. Consequently,

$$\frac{\partial f_k}{\partial p_j}(p) = \sum_{\substack{T \subseteq [K] \setminus \{j\} \\ |T| = k-1}} \left(\prod_{i \in T} p_i \right) \left(\prod_{\ell \notin T \cup \{j\}} (1 - p_\ell) \right) \geq 0. \tag{B.6}$$

This expression is a sum of non-negative terms, which shows that f_k is coordinate-wise non-decreasing. Dividing by $f_k(p) > 0$ then yields the corresponding non-negativity result for $\frac{\partial}{\partial p_j} \log f_k(p)$. □

Proposition B.1 (Monotonicity of the expectation of p_k with respect to λ)**.** *Let $y = (y_1, \dots, y_K)$ be the properties of interest, q_λ the guided reverse kernel at guidance strength λ, and Ω_t the candidate set of graphs at time step t. Define predictors $p_j : \Omega_t \to [0, 1]$, one for each property, and let $p_{rule}(p)$ denote the probability of a general rule as a function of the property predictors $p = (p_1, \dots, p_K)$. Suppose that (i) the predictors are not negatively correlated under q_λ, i.e. $\mathrm{Cov}_{q_\lambda}(p_j, p_k) \geq 0$ for all j, k, and (ii) the rule probability p_{rule} is coordinate-wise non-decreasing in each p_j. Then, for any fixed reverse step t and any $\lambda \geq 0$, the expectation of each property predictor satisfies*

$$\frac{d}{d\lambda} \mathbb{E}_{q_\lambda}[p_k] \geq 0, \tag{B.7}$$

i.e. $\mathbb{E}_{q_\lambda}[p_k]$ is non-decreasing in λ.

Proof. For any bounded function $h : \Omega^t \to \mathbb{R}$, differentiating its expectation under q_λ with respect to λ gives

$$\frac{d}{d\lambda} \mathbb{E}_{q_\lambda}[h] = \mathrm{Cov}_{q_\lambda}(h, u), \tag{B.8}$$

where u is the guidance utility.

By construction, the rule probability satisfies

$$\nabla_p \log p_{\text{rule}}(p) = (a_1, \dots, a_K), \qquad a_j = \frac{\partial}{\partial p_j} \log p_{\text{rule}}(p) \geq 0, \tag{B.9}$$

since p_{rule} is coordinate-wise non-decreasing. Hence, by the chain rule, the utility u can be written (to first order) as a non-negative linear combination of the predictors:

$$u \approx \sum_{j=1}^{K} a_j \, p_j. \tag{B.10}$$

Choosing $h = p_k$ then yields

$$\begin{aligned} \frac{d}{d\lambda} \mathbb{E}_{q_\lambda}[p_k] &\approx \mathrm{Cov}_{q_\lambda}\left(p_k, \sum_{j=1}^{K} a_j \, p_j \right) \\ &= \sum_{j=1}^{K} a_j \, \mathrm{Cov}_{q_\lambda}(p_k, p_j). \end{aligned} \tag{B.11}$$

By assumption, each covariance term is non-negative, and each $a_j \geq 0$, so the sum is non-negative. Formally, the terms discarded in the first-order approximation of u correspond to higher-order contributions in the local expansion around G^t. These contributions scale as $O(\|G^{t-1} - G^t\|^2)$; since the reverse step in diffusion is small, they are negligible to first order in the step size (and guidance strength) and therefore do not affect the non-negative sign of the leading term. Therefore we obtain the result in (B.7), which implies that $\mathbb{E}_{q_\lambda}[p_k]$ is non-decreasing in λ. □

Using Lemma B.1 and the same assumptions as in Proposition B.1, it follows that, when guiding for a cardinality rule, the tails of the expected probability of the graphs satisfying exactly r properties are monotonic. In addition, the expected probability of satisfying all K properties is monotonically non-decreasing.

Proposition B.2 (Tail monotonicity of the expected r-of-K probability)**.** *Under the same assumptions as in Lemma B.1, set $S = \sum_{i=1}^{K} X_i$ with mean $\mu_S(p) = \sum_{i=1}^{K} p_i$. For $1 \le r \le K-1$, define $p_r(p) := \mathbb{P}(S = r|p)$, the Poisson–binomial pmf at r. Furthermore, assume that $\sigma^2_{\mu_S} = \mathbb{V}_{q_\lambda}[\mu_S] < +\infty$. Then, under the same conditions as in Proposition B.1, for every r and any $c > 0$ we obtain:*

- $\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_r] \ge 0$ *whenever* $\mathbb{E}_{q_\lambda}[\mu_S] \le r - 1 - c\sigma_{\mu_S}$,
- $\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_r] \le 0$ *whenever* $\mathbb{E}_{q_\lambda}[\mu_S] \ge r + 1 + c\sigma_{\mu_S}$,

up to a tail probability of order $\frac{1}{c^2}$ by Chebyshev's inequality.

Proof. Fix $r \in \{1, \dots, K-1\}$. For $m \in \{1, \dots, K\}$, let $S_{-m} = \sum_{i \ne m} X_i$, and denote $p_j^{(-m)}(p_{-m}) = \mathbb{P}(S_{-m} = j \,|\, p_{-m})$. Conditioning on X_m yields

$$p_r(p) = p_m \, p_{r-1}^{(-m)}(p_{-m}) + (1 - p_m) \, p_r^{(-m)}(p_{-m}). \tag{B.12}$$

Differentiating gives

$$\frac{\partial p_r}{\partial p_m}(p) = p_{r-1}^{(-m)}(p_{-m}) - p_r^{(-m)}(p_{-m}). \tag{B.13}$$

By Darroch's theorem [8], there exists a mode

$$m^*_{-m}(p_{-m}) \in \{\lfloor \mu_S(p) - p_m \rfloor, \lceil \mu_S(p) - p_m \rceil\} \tag{B.14}$$

such that $p_j^{(-m)}(p_{-m})$ is increasing up to $j = m^*_{-m}$ and decreasing thereafter. Consequently:

- if $\mu_S(p) \le r - 1$, then $\frac{\partial p_r}{\partial p_m} \ge 0$ for all m, so p_r is coordinate-wise non-decreasing;
- if $\mu_S(p) \ge r + 1$, then $\frac{\partial p_r}{\partial p_m} \le 0$ for all m, so p_r is coordinate-wise non-increasing.

Now, differentiating the expectation of p_r under q_λ with respect to λ gives

$$\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_r] = \mathrm{Cov}_{q_\lambda}(p_r, u), \tag{B.15}$$

similarly to Equation (B.8). Using both a first-order Taylor approximation of p_r around $\mu_p = \mathbb{E}_{q_\lambda}[p]$, and the linearisation $\tilde{u}$ from Equation (B.10), we can approximate the covariance as

$$\mathrm{Cov}_{q_\lambda}(p_r, u) \approx \sum_{j=1}^{K} \frac{\partial p_r}{\partial p_j}(\mu_p) \, \mathrm{Cov}_{q_\lambda}(p_j, \tilde{u}). \tag{B.16}$$

Under the assumptions of Proposition B.1, we showed that $\mathrm{Cov}_{q_\lambda}(p_j, \tilde{u}) \ge 0$. Hence, to leading order, the sign of $\mathrm{Cov}_{q_\lambda}(p_r, u)$ is determined by the sign of each $\frac{\partial p_r}{\partial p_j}(\mu_p)$ in the summation. Since the diffusion process is considered in the small-variance regime, and the remainder of p_r scales as $O(\sigma^2_{\mu_S})$ whereas the linear term scales as $O(\sigma_{\mu_S})$, the former cannot

affect the sign of the covariance. Similarly, the interaction terms with the residual of the linear approximation of u contribute only higher-order covariances and therefore do not affect the sign.

By Chebyshev's inequality, for any $c > 0$ we have

$$\mathbb{P}\big(|\mu_S - \mathbb{E}_{q_\lambda}[\mu_S]| > c\sigma_{\mu_S}\big) \leq \frac{1}{c^2}, \tag{B.17}$$

where $\mathbb{E}_{q_\lambda}[\mu_S] = \sum_j \mu_j$. Thus, with probability at least $1 - \frac{1}{c^2}$, we distinguish two cases:

- if $\mathbb{E}_{q_\lambda}[\mu_S] \leq r - 1 - c\sigma_{\mu_S}$, then $\frac{\partial p_r}{\partial p_j}(\mu_p) \geq 0$ for all j, which implies $\mathrm{Cov}_{q_\lambda}(p_r, u) \geq 0$;
- if $\mathbb{E}_{q_\lambda}[\mu_S] \geq r + 1 + c\sigma_{\mu_S}$, then $\frac{\partial p_r}{\partial p_j}(\mu_p) \leq 0$ for all j, which implies $\mathrm{Cov}_{q_\lambda}(p_r, u) \leq 0$, proving the claim.

□

Proposition B.3 (Monotonicity of the expected K-of-K probability)**.** *Under the guided kernel q_λ, and the same assumptions as in Lemma B.1 and Proposition B.1, the expectation of $p_K(p) = \prod_{j=1}^K p_j$, which denotes the probability that all K conditions are satisfied, is non-decreasing in λ:*

$$\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_K] \geq 0. \tag{B.18}$$

Consequently, the maximum is attained in the limit $\lambda \to \infty$.

Proof. Taking the derivative of the function $\mathbb{E}_{q_\lambda}[p_K]$ with respect to λ gives

$$\frac{d}{d\lambda}\mathbb{E}_{q_\lambda}[p_K] = \mathrm{Cov}_{q_\lambda}(p_K, u), \tag{B.19}$$

as in Equation (B.8). The first-order Taylor expansion of p_K gives

$$\mathrm{Cov}_{q_\lambda}(p_K, u) \approx \sum_{j=1}^{K} \frac{\partial p_K}{\partial p_j}(\mu_p)\, \mathrm{Cov}_{q_\lambda}(p_j, \tilde{u}). \tag{B.20}$$

By the same assumptions of Proposition B.1, we have $\mathrm{Cov}_{q_\lambda}(p_j, \tilde{u}) \geq 0$ for all j. Furthermore, $\frac{\partial}{\partial p_j} p_K(\mu_p) = \prod_{\ell \neq j} \mu_\ell \geq 0$, so the leading term is non-negative. Since the remainder terms are higher order in σ_{μ_S} and thus negligible (c.f. Proposition B.2), we can conclude that $\mathrm{Cov}_{q_\lambda}(p_K, u) \geq 0$. □

Remark. Proposition B.3 establishes that the expected probability of satisfying all K properties is monotonically non-decreasing, but it does not guarantee that its limit equals 1.

C

ADDITIONAL EXPERIMENTAL RESULTS

C.1 PCA PROJECTIONS BY CLASSIFIER ARCHITECTURE

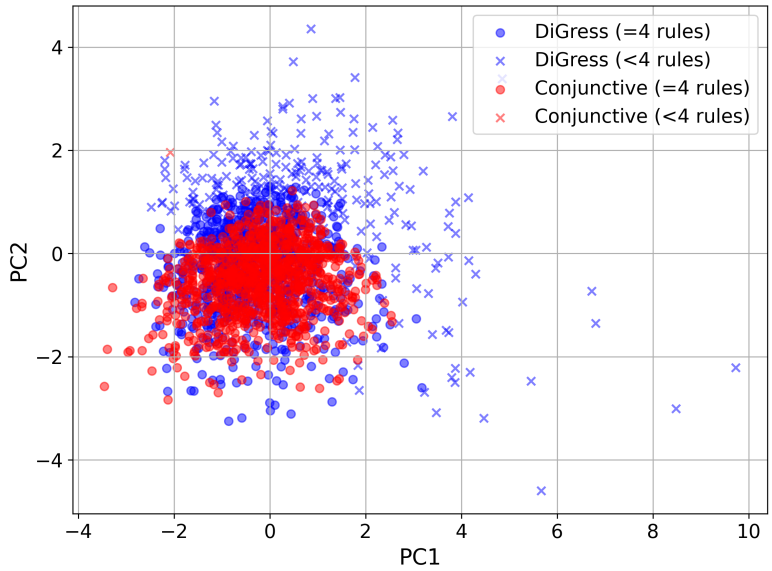
This appendix presents PCA projections of the generated molecules across classifier architectures at $\lambda = 47.82$ (see Figures 16 and 17), complementing the UMAP analyses reported in the main text. For both guidance rules, the PCA of shared-1L resembles independent-2L, while shared-2L appears denser and more restricted relative to the DiGress unconditional model.

C.2 EXTENDED BLACK-BOX ANALYSIS

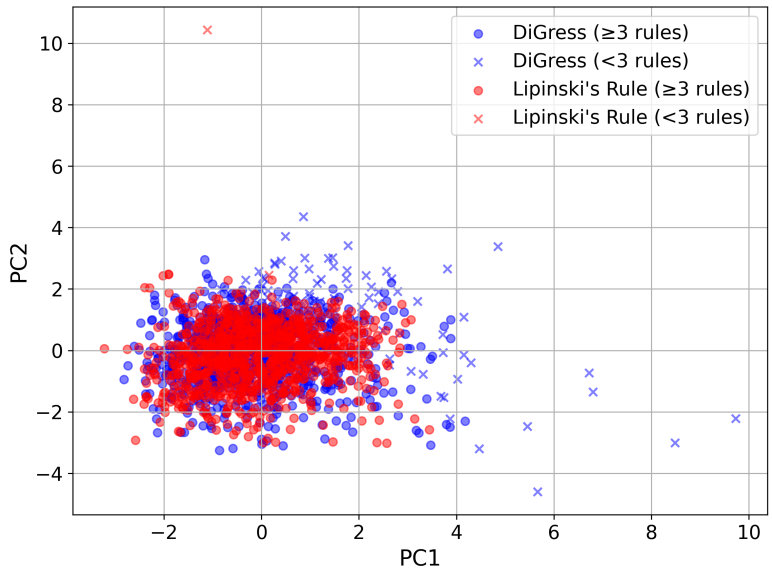
The trade-offs across compliance, validity, KL and FCD are shown in Figure 18 and discussed in the main text. Consistently, the UMAP and PCA projections ($\lambda = 47.82$) indicate that the black-box baseline closely resembles the behaviour of our logical method, though with reduced coverage of the property space under the conjunctive rule (Figures 19 and 20). These visualisations confirm that the baseline does not induce mode collapse, and that its external diversity remains comparable to that of our logical guidance.

C.3 ILLUSTRATIVE EXAMPLES OF GUIDED MOLECULES

Examples of molecules generated under Lipinski's rule guidance with the shared-1L classifier at guidance strength $\lambda = 47.82$ are shown in Figure 21. These samples illustrate the qualitative outcomes of the model without additional filtering or selection.

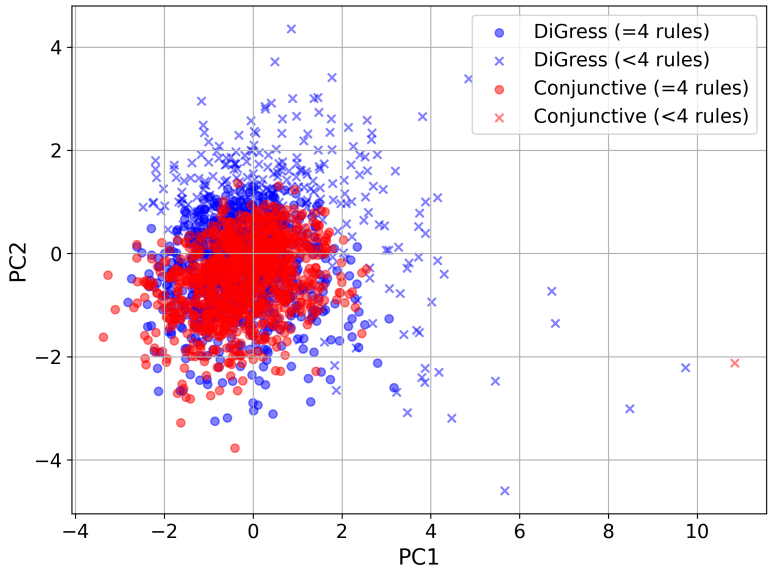


(a) Conjunctive guidance.

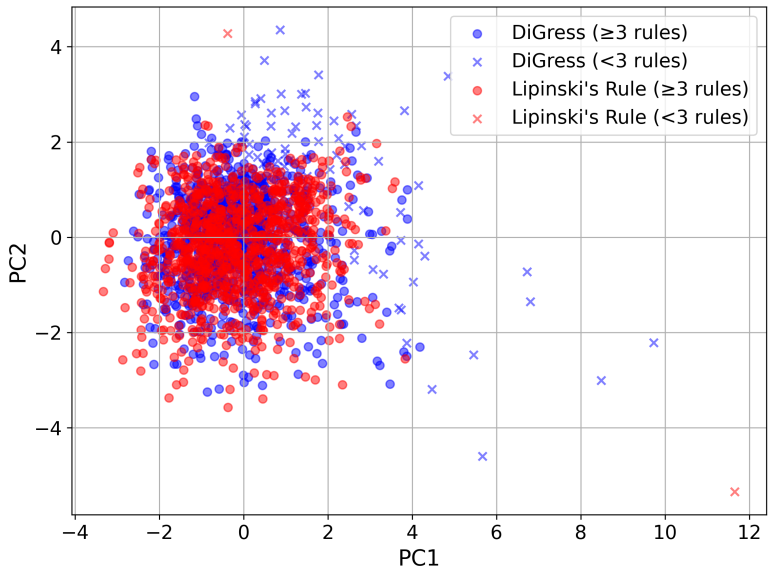


(b) Lipinski's rule guidance.

Figure 16: PCA projection of molecules from the DiGress model, overlaid with those generated using the shared-2L classifier. Both sets are projected onto the principal components learned from DiGress samples for consistency.

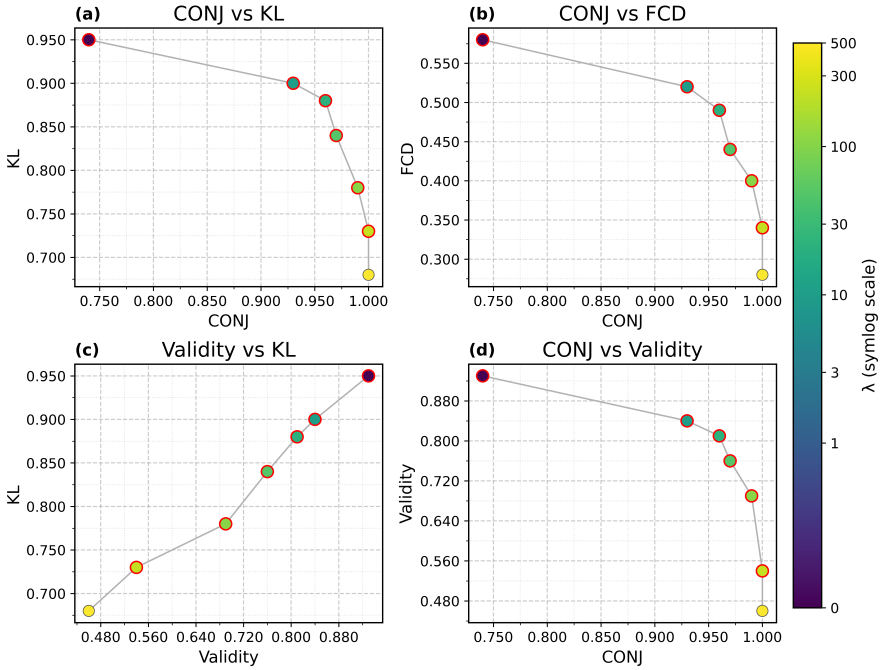


(a) Conjunctive guidance.

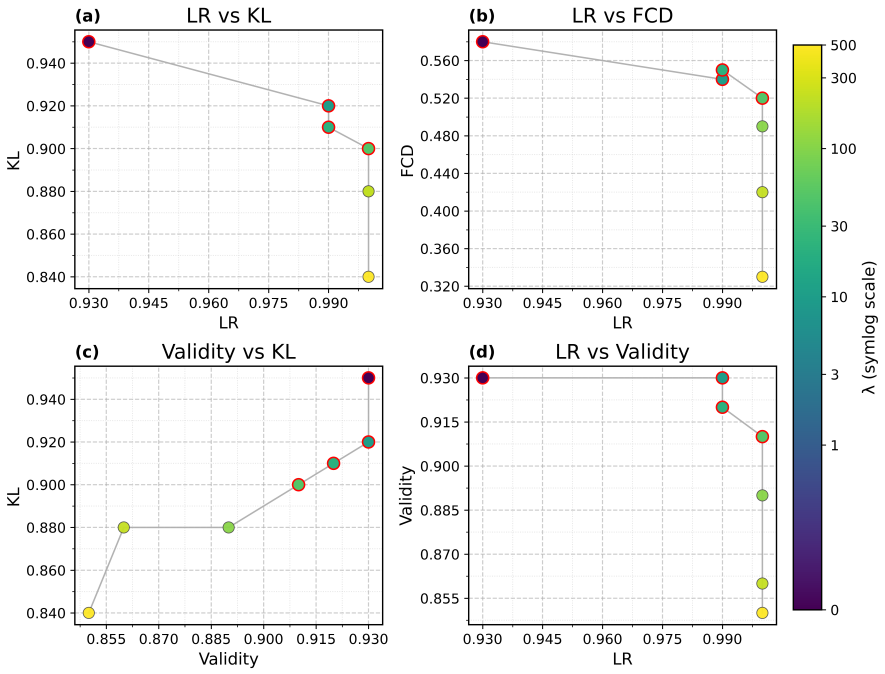


(b) Lipinski's rule guidance.

Figure 17: PCA projection of molecules from the DiGress model, overlaid with those generated using the shared-1L classifier. Both sets are projected onto the principal components learned from DiGress samples for consistency.

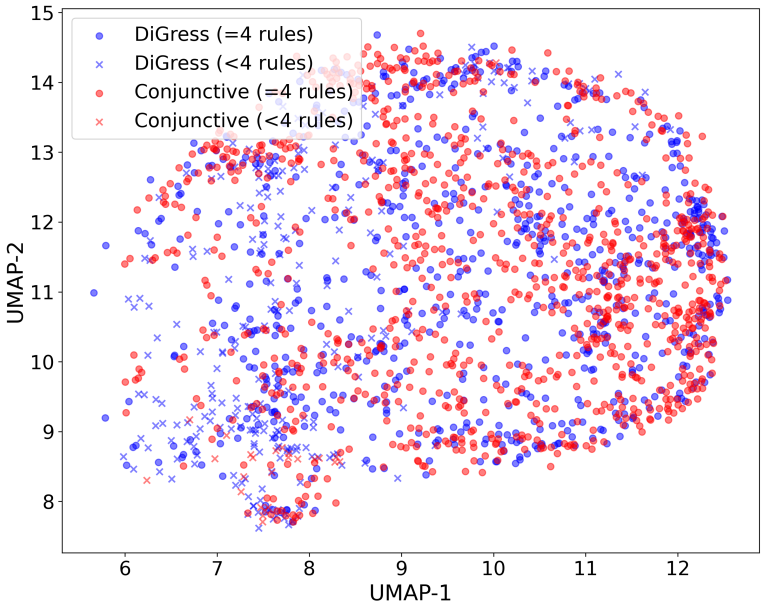


(a) Conjunctive guidance.

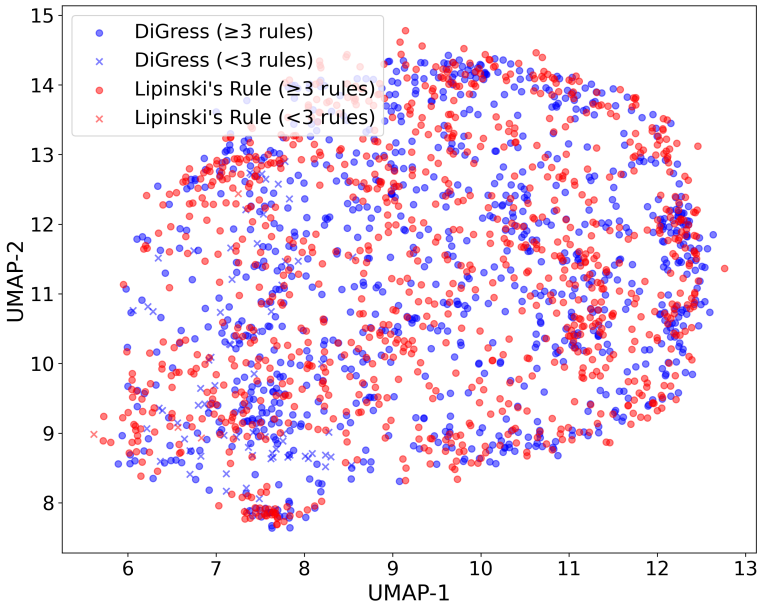


(b) Lipinski's rule guidance.

Figure 18: Trade-offs between compliance, distributional similarity (KL, FCD), and validity across guidance strengths λ (colour-coded, log scale) for black-box guidance. Red-circled points indicate the Pareto frontier.

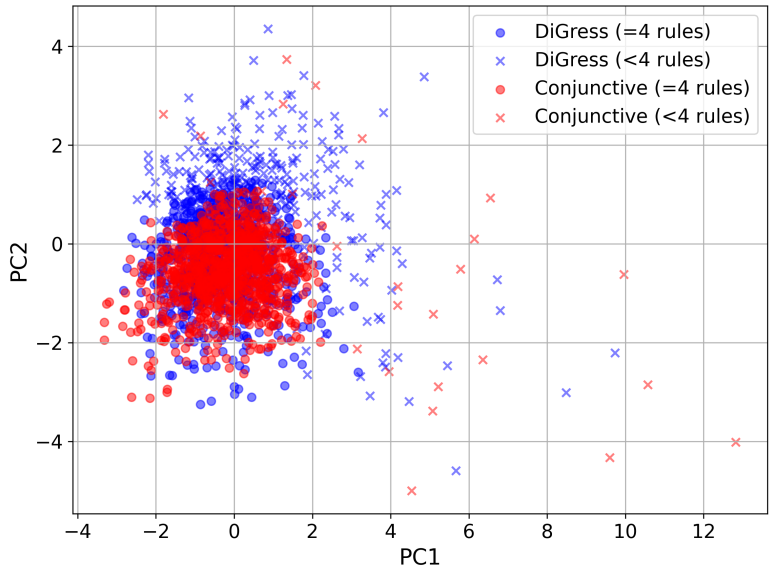


(a) Conjunctive guidance.

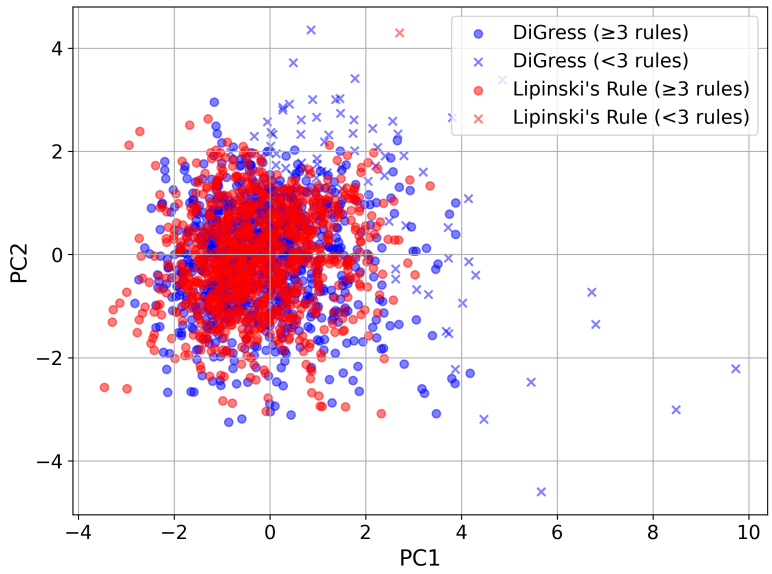


(b) Lipinski's rule guidance.

Figure 19: UMAP projection of molecules from the DiGress model, overlaid with those generated using the black-box classifier. All samples are projected onto the UMAP embedding learned from the training data for consistency.



(a) Conjunctive guidance.



(b) Lipinski's rule guidance.

Figure 20: PCA projection of molecules from the DiGress model, overlaid with those generated using the black-box classifier. Both sets are projected onto the principal components learned from DiGress samples for consistency.

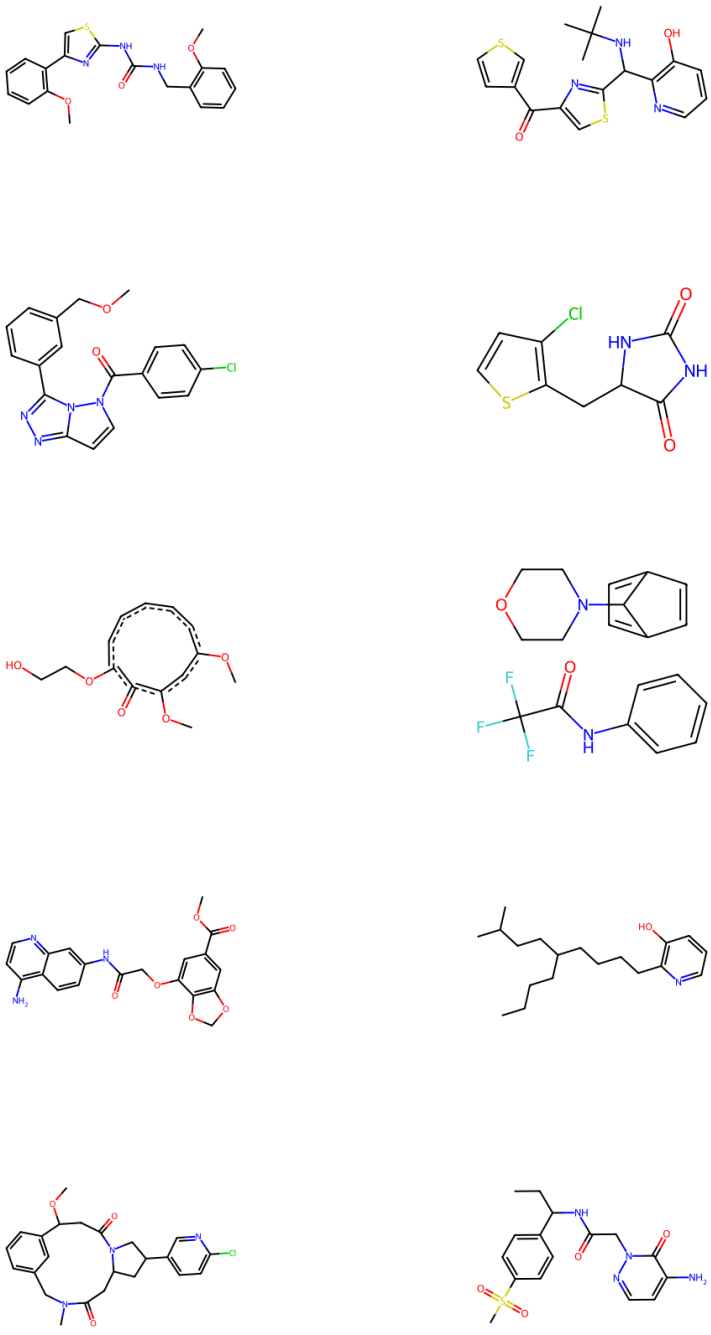


Figure 21: Uncurated examples of molecules generated under logical guidance for Lipinski's Rule.

BIBLIOGRAPHY

[1] Dániel Bajusz, Attila Rácz, and Károly Héberger. "Why is Tanimoto Index an Appropriate Choice for Fingerprint-based Similarity Calculations?" In: *Journal of Cheminformatics* 7.20 (2015). DOI: 10.1186/s13321-015-0069-3.

[2] Leslie Z. Benet et al. "BDDCS, the Rule of 5 and Drugability". In: *Advanced Drug Delivery Reviews* 101 (2016). DOI: 10.1016/j.addr.2016.05.007.

[3] Camille Bilodeau et al. "Generative Models for Molecular Discovery: Recent Advances and Challenges". In: *WIREs Computational Molecular Science* 12.5 (2022). DOI: 10.1002/wcms.1608.

[4] Nathan Brown et al. "GuacaMol: Benchmarking Models for de novo Molecular Design". In: *Journal of Chemical Information and Modeling* 59.3 (2019). DOI: 10.1021/acs.jcim.8b00839.

[5] Andrew Campbell et al. "A Continuous Time Framework for Discrete Denoising Models". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 2022, pp. 28266–28279.

[6] RDKit community. *RDKit: Open-source Cheminformatics*. Version 2023.03.2. URL: https://www.rdkit.org.

[7] Thomas M. Cover and Joy A. Thomas. "Maximum Entropy". In: *Elements of Information Theory*. John Wiley & Sons, 2005. Chap. 12, pp. 409–425. DOI: 10.1002/047174882X.ch12.

[8] John N. Darroch. "On the Distribution of the Number of Successes in Independent Trials". In: *Annals of Mathematical Statistics* 35 (1964). DOI: 10.1214/AOMS/1177703287.

[9] Nicola De Cao and Thomas Kipf. "MolGAN: An Implicit Generative Model for Small Molecular Graphs". In: *ICML 2018 Workshop on Theoretical Foundations and Applications of Deep Generative Models*. Vol. 80. PMLR, 2018.

[10] Zhiwei Deng et al. "Continuous Graph Flow". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 108. PMLR, 2020.

[11] Prafulla Dhariwal and Alex Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Proceedings of the 35th International Conference on Neural Information Processing Systems*. 2021, pp. 8780–8794.

[12] Vijay P. Dwivedi and Xavier Bresson. *A Generalization of Transformer Networks to Graphs*. 2021. DOI: 10.48550/arXiv.2012.09699.

[13] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[14] Elyas Goli et al. "ChemNet: a deep neural Network for Advanced Composites Manufacturing". In: *The Journal of Physical Chemistry B* 124 (42 2020). DOI: 10.1021/acs.jpcb.0c03328.

[15] Arthur Gretton et al. "A Kernel Two-sample Test". In: *The Journal of Machine Learning Research* 13 (2012). DOI: 10.5555/2188385.2188410.

[16] Kilian Konstantin Haefeli et al. *Diffusion Models for Graphs Benefit from Discrete State Spaces*. 2023. DOI: 10.48550/arXiv.2210.01549.

[17] William L. Hamilton. *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer, 2020. DOI: 10.1007/978-3-031-01588-5.

[18] John Healy and Leland McInnes. "UMAP: Uniform Manifold Approximation and Projection". In: *Nature Reviews Methods Primers* 4.82 (2024). DOI: 10.1038/s43586-024-00363-x.

[19] Weihua Hu et al. "OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs". In: *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. Vol. 35. 2021.

[20] Han Huang et al. "Conditional Diffusion Based on Discrete Graph Structures for Molecular Graph Generation". In: *Proceedings of the 37th AAAI Conference on Artificial Intelligence and 35th Conference on Innovative Applications of Artificial Intelligence and 13th Symposium on Educational Advances in Artificial Intelligence*. 2023, pp. 4302–4311. DOI: 10.1609/aaai.v37i4.25549.

[21] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. "Score-based Generative Modeling of Graphs via the System of Stochastic Differential Equations". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. PMLR, 2022.

[22] Bharti Khemani et al. "A Review of Graph Neural Networks: Concepts, Architectures, Techniques, Challenges, Datasets, Applications, and Future Directions". In: *Journal of Big Data* 11 (2024). DOI: 10.1186/s40537-023-00876-4.

[23] Thomas N. Kipf and Max Welling. *Variational Graph Auto-Encoders*. 2016. DOI: 10.48550/arXiv.1611.07308.

[24] Jathin Korrapati, Tanish Baranwal, and Rahul Shah. *Discrete vs. Continuous Trade-offs for Generative Models*. 2024. DOI: 10.48550/arXiv.2412.19114.

[25] Jure Leskovec. *Stanford CS224W: Machine Learning with Graphs*. Course slides, Stanford University, accessed October 2025. URL: https://web.stanford.edu/class/cs224w/.

[26] Renjie Liao et al. "Efficient Graph Generation with Graph Recurrent Attention Networks". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 2019, pp. 4255–4265.

[27] Christopher A. Lipinski. "Drug-like Properties and the Causes of Poor Solubility and Poor Permeability". In: *Journal of Pharmacological and Toxicological Methods* 44.1 (2000). DOI: 10.1016/S1056-8719(00)00107-6.

[28] Chengyi Liu et al. "Generative Diffusion Models on Graphs: Methods and Applications". In: *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*. 2023, pp. 6702–6711. DOI: 10.24963/ijcai.2023/751.

[29] Gang Liu et al. "Graph Diffusion Transformers for Multi-conditional Molecular Generation". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. 2025, pp. 8065–8092.

[30] Jenny Liu et al. "Graph Normalizing Flows". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 2019, pp. 13578–13588.

[31] Dominic Masters et al. *GPS++: An Optimised Hybrid MPNN/Transformer for Molecular Property Prediction*. 2022. DOI: 10.48550/arXiv.2212.02229.

[32] Grzegorz Miebs et al. "Beyond the Arbitrariness of Drug-likeness Rules: Rough Set Theory and Decision Rules in the Service of Drug Design". In: *Applied Sciences* 14.21 (2024). DOI: 10.3390/app14219966.

[33] Luis Müller et al. "Attending to Graph Transformers". In: *Transactions on Machine Learning Research* (2024).

[34] Matteo Ninniri, Marco Podda, and Davide Bacciu. "Classifier-free Graph Diffusion for Molecular Property Targeting". In: *Machine Learning and Knowledge Discovery in Databases. Research Track. ECML PKDD 2024*. Vol. 14944. Lecture Notes in Computer Science. Springer, 2024. DOI: 10.1007/978-3-031-70359-1_19.

[35] Chenhao Niu et al. "Permutation Invariant Graph Generation via Score-based Generative Modeling". In: *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 108. PMLR, 2020.

[36] Ethan Perez et al. "FiLM: Visual Reasoning with a General Conditioning Layer". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*. 2018, pp. 3942–3951. DOI: 10.1609/aaai.v32i1.11671.

[37] Daniil Polykovskiy et al. "Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models". In: *Frontiers in Pharmacology* 11 (2020). DOI: 10.3389/fphar.2020.565644.

[38] Raghunathan Ramakrishnan et al. "Quantum Chemistry Structures and Properties of 134 Kilo Molecules". In: *Scientific Data* 1.1 (2014). DOI: 10.1038/sdata.2014.22.

[39] Yinuo Ren et al. "How Discrete and Continuous Diffusion Meet: Comprehensive Analysis of Discrete Diffusion Models via a Stochastic Integral Framework". In: *The 13th International Conference on Learning Representations*. 2025.

[40] Kartik Sharma, Srijan Kumar, and Rakshit S. Trivedi. "Diffuse, Sample, Project: Plug-and-Play Controllable Graph Generation". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. PMLR, 2024, pp. 44545–44564.

[41] Martin Simonovsky and Nikos Komodakis. "GraphVAE: Towards Generation of Small Graphs using Variational Autoencoders". In: *International Conference on Artificial Neural Networks (ICANN-2018)*. Vol. 11139. Lecture Notes in Computer Science. Springer, 2018. DOI: 10.1007/978-3-030-01418-6_41.

[42] Yuxuan Song et al. "Smooth Interpolation for Improved Discrete Graph Generative Models". In: *Proceedings of the 42nd International Conference on Machine Learning*. 2025.

[43] Xiangru Tang et al. "A Survey of Generative AI for de novo Drug Design: New Frontiers in Molecule and Protein Generation". In: *Briefings in Bioinformatics* 25.4 (2024). DOI: 10.1093/bib/bbae338.

[44] Victor M. Tenorio et al. *Graph Guided Diffusion: Unified Guidance for Conditional Graph Generation*. 2025. DOI: 10.48550/arXiv.2505.19685.

[45] Clement Vignac et al. "DiGress: Discrete Denoising Diffusion for Graph Generation". In: *The 11th International Conference on Learning Representations*. 2023.

[46] Liang Wang et al. *Diffusion Models for Molecules: A Survey of Methods and Tasks*. 2025. DOI: 10.48550/arXiv.2502.09511.

[47] David Weininger. "SMILES, A Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules". In: *Journal of Chemical Information and Computer Sciences* 28.1 (1988). DOI: 10.1021/ci00057a005.

[48] David Weininger. "SMILES. 3. DEPICT. Graphical Depiction of Chemical Structures". In: *Journal of Chemical Information and Computer Sciences* 30.3 (1990). DOI: 10.1021/ci00067a005.

[49] David Weininger, Arthur Weininger, and Joseph L. Weininger. "SMILES. 2. Algorithm for Generation of Unique SMILES Notation". In: *Journal of Chemical Information and Computer Sciences* 29.2 (1989). DOI: 10.1021/ci00062a008.

[50] Robin Winter et al. "Learning Continuous and Data-driven Molecular Descriptors by Translating Equivalent Chemical Representations". In: *Chemical Science* 10 (2019). DOI: 10.1039/C8SC04175J.

[51] Zhe Xu et al. "Discrete-state Continuous-time Diffusion for Graph Generation". In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. 2025, pp. 79704–79740.

[52] Mengchun Zhang et al. *A Survey on Graph Diffusion Models: Generative AI in Science for Molecule, Protein and Material*. 2023. DOI: 10.48550/arXiv.2304.01565.

[53] Yanqiao Zhu et al. "A Survey on Deep Graph Generation: Methods and Applications". In: *Proceedings of the First Learning on Graphs Conference (LoG 2022)*. Vol. 198. PMLR, 2022.