
MOLGUIDANCE: ADVANCED GUIDANCE STRATEGIES FOR CONDITIONAL MOLECULAR GENERATION WITH FLOW MATCHING

Jirui Jin^{1,2†}, Cheng Zeng^{1,2†*}, Pawan Prakash^{2,3}, Ellad B. Tadmor⁴, Adrian Roitberg^{1,2}, Richard G. Hennig^{2,5}, Stefano Martiniani^{6,7,8,9*}, and Mingjie Liu^{1,2*}

[†]These authors contribute equally.

¹Department of Chemistry, University of Florida, Gainesville, FL 32611, USA

²Quantum Theory Project, University of Florida, Gainesville, FL 32611, USA

³Department of Physics, University of Florida, Gainesville, FL 32611, USA

⁴Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN 55455, USA

⁵Department of Materials Science & Engineering, University of Florida, Gainesville, FL 32611, USA

⁶Center for Soft Matter Research, Department of Physics, New York University, New York 10003, USA

⁷Simons Center for Computational Physical Chemistry, Department of Chemistry, New York University, New York 10003, USA

⁸Courant Institute of Mathematical Sciences, New York University, New York 10003, USA

⁹Center for Neural Science, New York University, New York 10003, USA

December 13, 2025

ABSTRACT

Key objectives in conditional molecular generation include ensuring chemical validity, aligning generated molecules with target properties, promoting structural diversity, and enabling efficient sampling for discovery. Recent advances in computer vision introduced a range of new guidance strategies for generative models, many of which can be adapted to support these goals. In this work, we integrate state-of-the-art guidance methods—including classifier-free guidance, autoguidance, and model guidance—in a leading molecule generation framework built on an SE(3)-equivariant flow matching process. We propose a hybrid guidance strategy that separately guides continuous and discrete molecular modalities—operating on velocity fields and predicted logits, respectively—while jointly optimizing their guidance scales via Bayesian optimization. Our implementation, benchmarked on the QM9 and QM14S datasets, achieves new state-of-the-art performance in property alignment for *de novo* molecular generation. The generated molecules also exhibit high structural validity. Furthermore, we systematically compare the strengths and limitations of various guidance methods, offering insights into their broader applicability.

Keywords Molecule Generation · Equivariant Flow Matching · Classifier-Free Guidance · Autoguidance · Model Guidance

Introduction

The generation of novel molecular structures with desired properties is crucial for drug design and chemical discovery [1, 2]. Traditional molecular design using high-throughput screening and structure-property relationship-based rational design relies on testing or iteratively modifying known compounds, but it probes only a minute fraction of the 10^{60} chemical space and is slow, costly, and biased by human intuition. [3, 4] In comparison, deep learning-based generative

*c.zeng@ufl.edu, sm7683@nyu.edu, mingjieliu@ufl.edu

models have shown immense promise in narrowing down search space and accelerating molecule discovery. [5, 6] These models learn a probabilistic representation of the vast chemical space and then directly sample molecules with desired properties. A key challenge in this domain is to effectively guide the generation process towards molecules that are valid, novel, and in satisfactory alignment with property constraints. This process is also referred to as property-guided generation.

The most straightforward approach of property-guided generation is vanilla conditional generation, which takes the property as the condition and molecule representation (*e.g.*, a molecular graph) as joint inputs, and learns a shared representation for properties and molecular structures that drives the sampling. [6, 7] While vanilla conditional generation proves simple and effective, it often struggles with precise property control—generated molecules may exhibit weak alignment with target properties or deviate from desired values. To address these limitations, guidance techniques have been developed to bias the sampling trajectory toward outputs that more closely satisfy the desired conditions. The core principle underlying guidance methods is to enhance conditional generation by steering the model toward property-specific molecular distributions. Various guidance methods have been proposed to achieve this goal. Seminal guidance methods range from classifier guidance [8], to classifier-free methods (CFG) [9, 10, 11] which aim to improve generation quality with better property alignment. More recent approaches, such as autoguidance (AG) [12] and model guidance (MG) [13] further seek to balance property adherence with sample diversity and computational efficiency. Given the proliferation of these guidance strategies, a systematic comparison is crucial for researchers and practitioners to understand their relative strengths, weaknesses, and suitability for conditional molecule generation.

A central challenge in molecular generation lies in developing effective strategies for applying guidance methods to multi-modality 3D molecular objects, such as discrete atom types and continuous atom positions. Although recent studies have begun exploring guidance mechanisms for molecular generation, existing approaches do not fully address this multi-modality. For instance, Luo et al. [14] applied classifier-free guidance to molecule generation on the QM9 dataset [15], but their approach did not address guidance for discrete molecular attributes such as atom types. By contrast, Nisonoff et al. [11] developed discrete guidance methods for SMILES [16] generation, but SMILES representations lack the atom coordinate information needed for 3D conformer generation. These gaps motivate the need for a comprehensive evaluation of guidance methods that can handle both continuous and discrete molecular modalities within a unified framework.

This gap motivates our development of a unified framework—MolGuidance that systematically handles both continuous and discrete molecular modalities. Moreover, building upon Nisonoff’s theoretical foundation [11], we conduct the first comprehensive investigation of discrete guidance formats, revealing fundamental differences between various mathematical formulations and their practical implications for molecular property control.

In a first kind of implementation, we adopted advanced property guidance methods in the context of *de novo* molecular generation based on the foundation of our flow matching model PropMolFlow (PMF) [17]. We aim to evaluate and benchmark four methods for conditional generation—including the vanilla conditional generation, CFG, AG, and MG—focusing on their ability to generate molecules that meet target property profiles, exhibit high structural validity and diversity, and require less computational overhead. Through this comparative study, we demonstrate the strengths and limitations of each guidance method while simultaneously conducting the first systematic investigation of discrete guidance formats within the CFG framework. Our contributions include discovering that empirically-motivated linear guidance on probabilities outperforms theoretically-derived logarithmic guidance on rate matrices due to stability considerations, and revealing that discrete molecular features (atom types, charges, bonds) exert greater control over molecular properties of generated molecules than continuous geometric arrangements (atomic positions) despite the latter receiving dominant training emphasis. We introduce hybrid guidance strategies that separately optimize continuous and discrete molecular modalities through Bayesian optimization, achieving state-of-the-art performance across guidance methods. Our comprehensive evaluation demonstrates that CFG delivers superior property alignment, AG provides optimal performance balance, while MG struggles in multi-modal molecular settings despite its success in computer vision tasks. We establish benchmarks across both QM9 (5 elements) and QM14S [18] (14 elements) datasets with DFT validation, demonstrating robustness and chemical relevance across different guidance methods and different datasets.

Results

Overview of MolGuidance

Flow matching has demonstrated notable success in unconditional molecule generation [19, 20], establishing itself as a powerful framework for learning complex molecular distributions and generating chemically valid structures. However, the integration of advanced guidance methods for conditional generation within flow matching frameworks remains largely unexplored, particularly for property-guided molecular design. We systematically evaluate three distinct

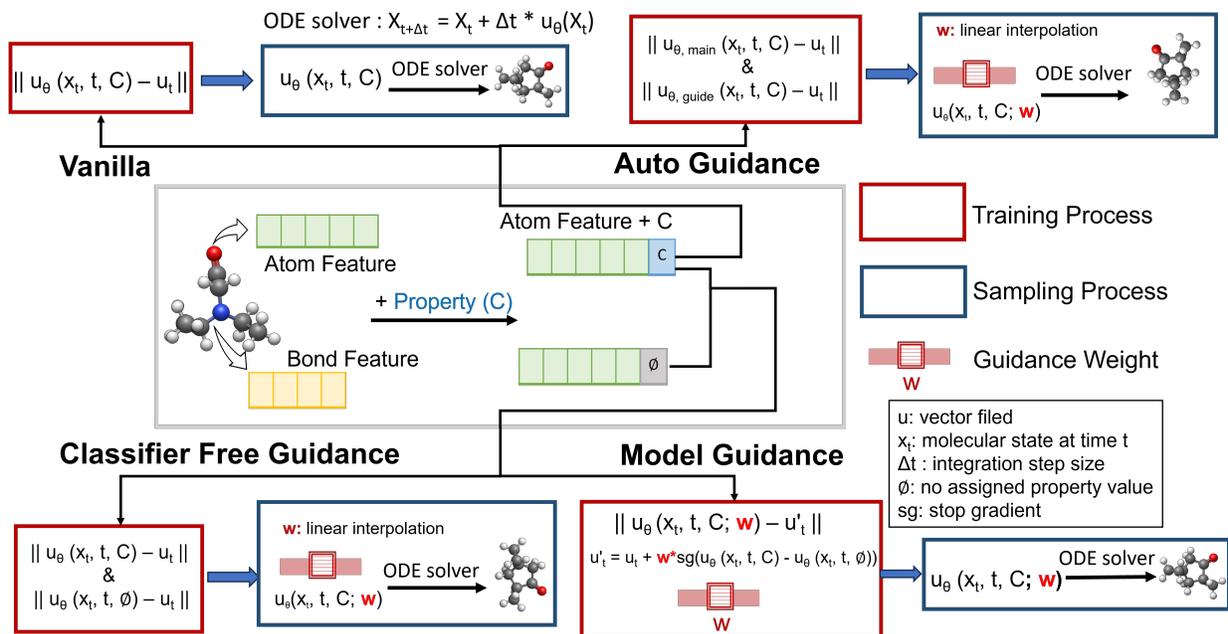


Figure 1: Overview of guidance methods for flow matching-based molecular generation.

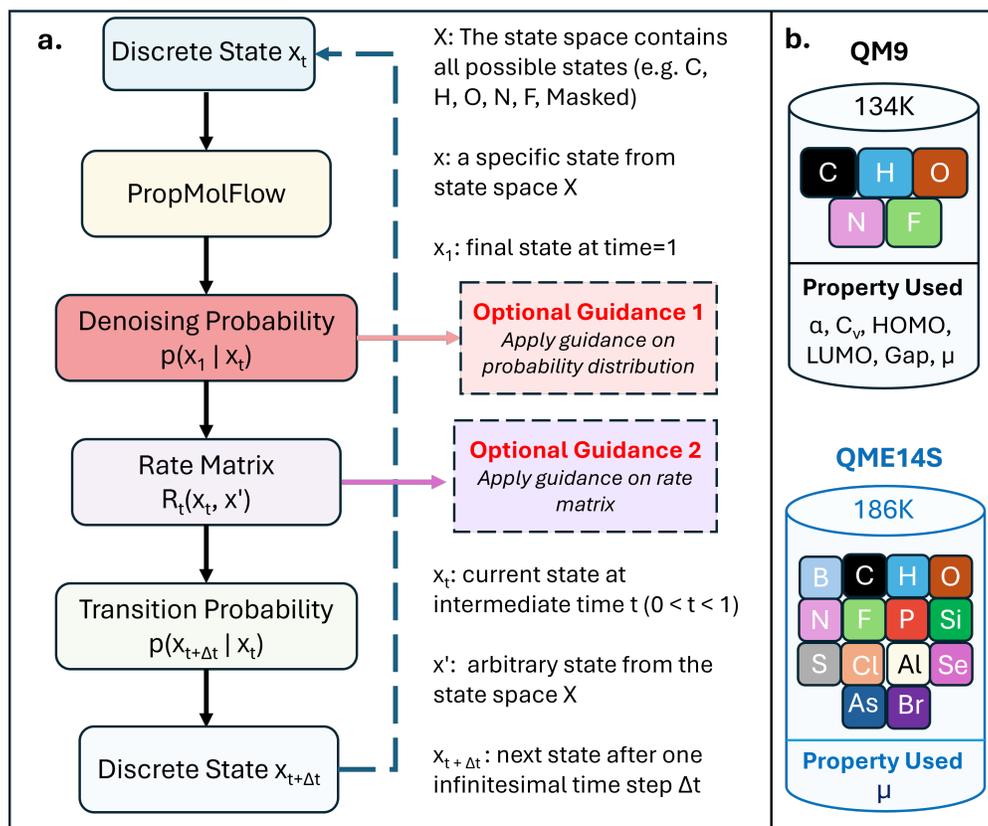


Figure 2: **a.** Sampling protocol for discrete molecular attributes (atoms, charges, and bonds) in the continuous-time Markov chain (CTMC) process, showing guidance can be applied either to rate matrices or probability distributions. **b.** Two benchmark datasets—QM9 and QME14S—used to evaluate different guidance methods, differing primarily in number of elements, dataset size, and target properties.

guidance strategies—CFG, AG, and MG—integrated within our previous conditional flow matching architecture PMF. As illustrated in Figure 1, our baseline vanilla model learns to transform Gaussian noise into molecular structures conditioned on properties through a flow-based generative process.

The core principle underlying guidance methods is to enhance conditional generation by steering the model toward property-specific molecular distributions, concentrating sampling in targeted regions of chemical space rather than sampling broadly from the unconditional distribution. Each guidance approach achieves this through fundamentally different mechanisms (Figure 1): Classifier-free guidance trains both conditional and unconditional models simultaneously, enabling guided sampling through weighted interpolation between conditional and unconditional predictions. The guidance weight (w) acts as a tunable parameter controlling the strength of property adherence during inference. Autoguidance employs a deliberately undertrained "inferior model"—with reduced network capacity and/or training epoch—that provides corrective signals to guide the vanilla model toward the target conditional distribution during sampling. In contrast, model guidance integrates guidance directly into the training objective through dynamic interaction between guidance signals and the learning process, eliminating the need for separate guidance computations during inference. The details of each guidance method are further discussed in the Methods Sections.

While guidance methods can be readily applied to the velocity fields of continuous data by linear interpolation (as commonly done in image generation and molecular coordinate generation), certain molecular modalities—such as atom types, charges, and bond order—requires its own approach to respect their discrete nature. State-of-the-art methods employ continuous-time Markov chain (CTMC) processes [21] to effectively model discrete variables. In this framework, illustrated in Figure 2a, the generative dynamics are governed by a rate matrix [11], which specifies the instantaneous transition rates between discrete states. At each time step t of the sampling process, the current noisy state x_t is passed through the PropMolFlow model to predict the denoising probability distribution $p(x_1|x_t)$, which reflects the model’s inferred distribution over the final, clean molecular state. This probability is then used to compute the rate matrix R_t , from which the transition probability $p(x_{t+\Delta t}|x_t)$ is derived for sampling the next state. This two-step process for handling discrete variables offers two distinct strategies for applying guidance ('Optional Guidance 1' and 'Optional Guidance 2'). The first approach applies guidance directly to the rate matrix by computing separate conditional and unconditional rate matrices and interpolating between them, thereby blending the dynamics of the two models. This method is theoretically grounded and follows the formulation proposed by Nisonoff et al. [11]. In contrast, the second, more empirical approach applies guidance to the denoising probability distribution prior to deriving the rate matrix. Specifically, it interpolates the conditional and unconditional probability predictions to obtain a guided distribution, from which a single, final rate matrix is subsequently computed. Furthermore, both approaches support either linear or logarithmic interpolation for combining the conditional and unconditional components. The linear scheme performs a direct, additive blend in probability space, while the logarithmic scheme operates in log-probability space to create a multiplicative blend of the guidance components. (Details are in

In this work, we employed two molecular datasets of increasing complexity to evaluate our model (Figure 2b). The first is the corrected rQM9 SDF dataset available at [22], which contains 134K molecules with explicit bond orders and atomic charges across five elements (C, H, O, N, F). We generated molecules by conditioning on six quantum-mechanical properties, including polarizability (α), heat capacity (C_v), ϵ_{HOMO} , ϵ_{LUMO} , $\Delta\epsilon$, and dipole moment (μ), where the detailed definitions are provided in

Before comparing different property-guided methods, we first systematically investigated optimal strategies for applying guidance to discrete and continuous variables. For discrete variables, we examined two key design choices—applying guidance in the probability-distribution space or directly in the rate-matrix space—and evaluated both linear and logarithmic interpolation schemes to identify the most stable and effective formulation. We then analyzed the guidance mechanism across both discrete and continuous modalities to determine which domain exerts greater influence on target property control. Building on these insights, we implemented a hybrid guidance strategy and employed Bayesian optimization to identify the optimal guidance weights for each method. Finally, using the established optimal protocol, we comprehensively benchmarked and compared four approaches—Vanilla, CFG, AG, and MG—for property-guided molecular generation, evaluating their performance in terms of property alignment, structural validity, and chemical diversity.

Engineering Guidance Formats for Stable Generation

We first explored the optimal format for applying guidance within our hybrid framework. Our strategy employs separate guidance weights for the continuous and discrete molecular modalities. In the following equations, we use CFG as the illustrative guidance method, though the formulation generalizes naturally to other guidance schemes. For the continuous modality (atomic positions), we apply a standard linear interpolation to the velocity fields (u), governed by the guidance weight w_1 :

$$u_{\text{guided}} = (1 - w_1) u_{\text{unconditional}} + w_1 u_{\text{conditional}}, \quad (1)$$

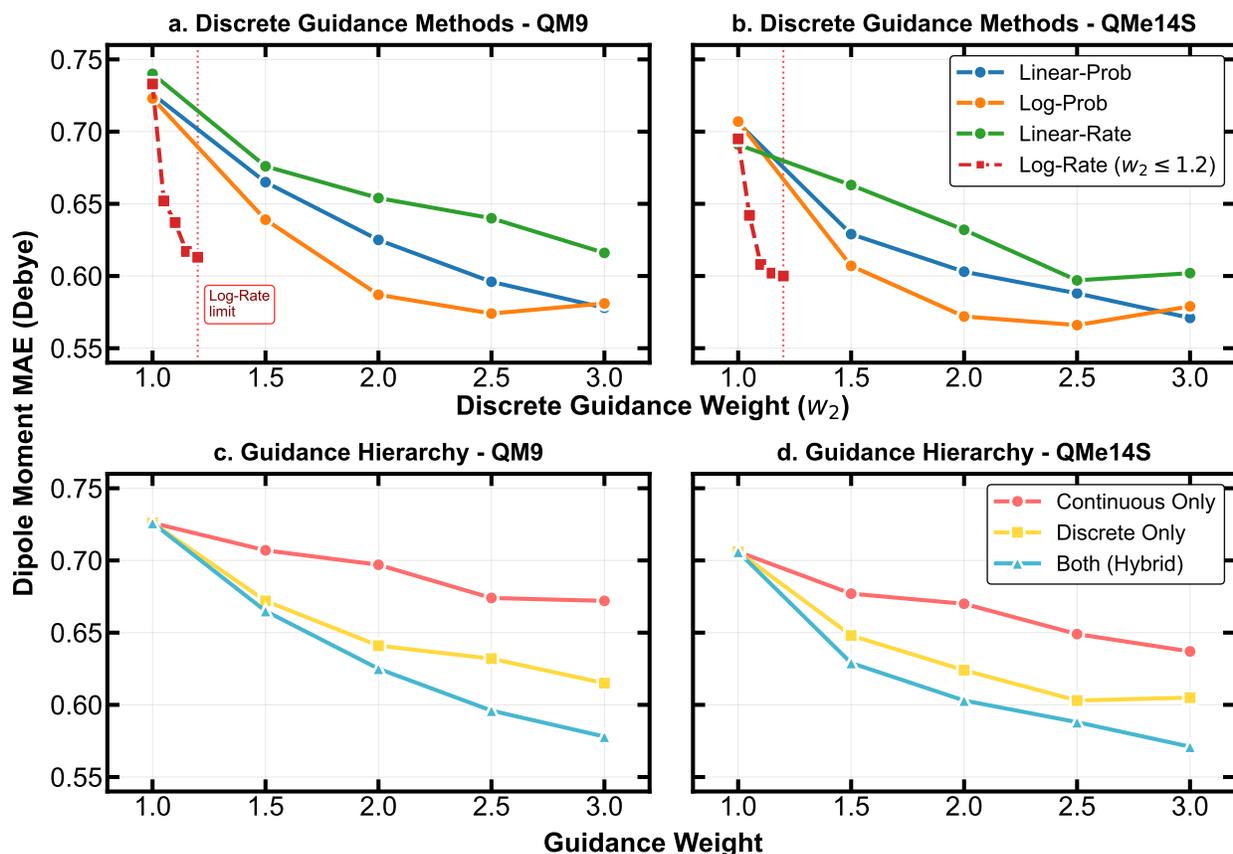


Figure 3: Comparison of guidance methods for dipole moment (μ) prediction across QM9 and QMe14S molecular datasets. Performance is measured as MAE in Debye units as a function of guidance weight. (a-b) Discrete guidance format comparison: Four methods are evaluated—linear guidance on probability (Linear-Prob), logarithm guidance on probability (Log-Prob), linear guidance on rate matrix (Linear-Rate), and logarithm guidance on rate matrix (Log-Rate). The Log-Rate method is constrained to $w_2 \leq 1.2$ because molecular validities drop rapidly at higher guidance weights, indicated by the vertical dotted line. (c-d) Guidance hierarchy comparison: Three strategies are compared—continuous-only guidance (atomic positions), discrete-only guidance (atom types, charges, bonds), and hybrid guidance (both domains). All methods demonstrate improved performance with increasing guidance weight, with optimal values typically occurring between $w = 2.0$ and 3.0 for numerically stable methods. Lower MAE values indicate better performance and alignment between target and generated molecule properties.

For the discrete modalities (atom types, charges, and bonds), we systematically investigate four distinct guidance formats, each governed by a separate guidance weight w_2 . These formats, which represent linear and logarithmic interpolation applied to either denoising probabilities $p_{1|t}$ or rate matrices $R_t(x_t, x')$, are defined as follows:

$$p_{\text{guided}} = (1 - w_2) p_{\text{unconditional}} + w_2 p_{\text{conditional}}, \quad (2)$$

$$p_{\text{guided}} = \exp\left[(1 - w_2) \log p_{\text{unconditional}} + w_2 \log p_{\text{conditional}}\right], \quad (3)$$

$$R_{\text{guided}} = (1 - w_2) R_{\text{unconditional}} + w_2 R_{\text{conditional}}, \quad (4)$$

$$R_{\text{guided}} = \exp\left[(1 - w_2) \log R_{\text{unconditional}} + w_2 \log R_{\text{conditional}}\right], \quad (5)$$

Figure 3a and 3b presents representative results for μ on both QM9 and QMe14S datasets, revealing substantial differences in guidance behavior across discrete weight ranges. In these experiments, we systematically varied both the continuous weight (w_1) and discrete weight (w_2) from 1.0 to 3.0 in increments of 0.5 for three of the four methods. The fourth method, logarithmic guidance on rate matrices, exhibited severe performance drop in molecular structural validity that constrained its discrete weight (w_2) exploration to a narrow range of 1.0 to 1.2 with increments of 0.05. The y-axis in Figure 3 represents the mean absolute error (MAE) between target μ values used for conditional generation and the μ values predicted by a pretrained regressor on the generated molecules, where lower values indicate better property alignment.

The logarithmic format on rate matrices, while theoretically derived by Nisonoff’s work [11], exhibits severe instability at discrete guidance weights beyond $w_2 = 1.2$, fundamentally limiting its practical applicability. In contrast, the logarithmic guidance on probabilities find its minimum property MAE around $w_2 = 2.5$ as shown in Figure 3a and 3b, achieving the best property alignment while maintaining numerical stability across the full weight range. Linear guidance formats show more gradual improvements with increasing weights but require higher guidance strengths to achieve comparable performance. We focus this analysis on CFG because AG shows low molecular structural validity with discrete guidance weights exceeding 2.0, while guidance-aware MG employs a guidance weight embedding, hence lacking in explicit flexibility for tuning guidance weights during sampling, making equivalent comparison infeasible. These findings highlight that empirically-motivated guidance formulations sometimes outperform theoretically-derived approaches in practical molecular generation, suggesting that numerical stability and guidance flexibility are critical considerations beyond mathematical elegance. The same trends are also observed on the other five properties of the QM9 dataset (Figure

Hybrid Guidance is Better than Unimodal Guidance

In PropMolFlow, the flow matching framework, the hierarchical loss weights were assigned during training, with atomic position receiving the highest weight, followed by bond order, formal charge, and atom type (see Supporting Information

This chemical hierarchy raises a critical question for hybrid guidance strategies: *Does guidance applied to the continuous space (atomic positions) exert a similarly dominant effect on property alignment compared to guidance on discrete variables?* To investigate this question systematically, we conduct controlled experiments comparing the relative impact of continuous versus discrete guidance weights on molecular property alignment. Given our previous finding that logarithmic formats amplify guidance effects but suffer from instability at higher discrete guidance weights, we employ linear guidance on probabilities for discrete components to ensure fair comparison with the inherently linear continuous guidance formulation.

As shown in Figure 3c and 3d, hybrid guidance consistently achieves superior property alignment in both data sets for μ , validating our hypothesis that combining continuous and discrete guidance takes advantage of the full chemical hierarchy. However, a counterintuitive finding emerges: discrete-only guidance (set no guidance on atomic positions) substantially outperforms continuous-only guidance (set no guidance on atom type, formal charge and bond order), contradicting the training loss hierarchy where atomic positions receive the dominant weight (see Supporting Information

The same trend has been observed across the other five properties in the QM9 dataset (Figure

Bayesian-optimized Hybrid Guidance Pushes the State-of-the-art Performance for Property Alignment

To identify the optimal guidance weights that offer the best alignment of generated molecules with target properties, we employed Bayesian optimization over the joint guidance weights for atomic positions (w_1) and discrete variables (w_2).

The logarithmic probability is used because it demonstrates good convergence and stability with increasing guidance weights. Implementation details of Bayesian optimization are included in the Bayesian Optimization of Guidance Weights Section. Using CFG conditioning on μ as the example (Figure

Our quantitative comparison of conditional generation across different guidance approaches and the three diffusion-based baseline models GeoLDM [7], GCDM [24], and JODO [25] shows that these guidance methods further boost the model’s performance. Firstly, we analyzed the property alignment performance by comparing PMF guidance methods to three baseline models on polarizability (α), HOMO-LUMO gap ($\Delta\epsilon$) and heat capacity (C_v) (Figure 4a). We note that the bond order was not used in three baseline models. Results for the remaining three properties can be found in Table

Additionally, we establish new benchmarks for all guidance methods on the more challenging QMe14S dataset using μ as the target property (Figure 4c). Interestingly, the property MAEs on the QMe14S dataset are generally lower than that on QM9 across all guidance methods. It demonstrates the robustness of our guidance framework across different molecular datasets and chemical space, with all methods maintaining consistent relative performance rankings despite the expanded elemental diversity (5 vs. 14 elements for respective QM9 and QMe14S) and larger molecular size distribution in QMe14S. Also, the trend of relative differences across guidance are kept for the QMe14S, revealing intrinsic, dataset-independent variation across guidance methods.

Since the property predictor shares the same model architecture with the PropMolFlow generative model, it may exhibit an inductive bias when predicting on molecules generated by PropMolFlow. To further confirm the performance of property alignment using PMF-CFG, we performed density functional theory (DFT) calculations for 500 molecules selected from the 10k molecules generated at the same level of theory using Gaussian [26] as the training data: B3LYP/6-31G(2df,p) for QM9 and B3LYP/TZVP for QMe14S, respectively. Single-point DFT calculations were conducted on the directly generated molecules for all properties except C_v , for which DFT properties of the relaxed molecules were used because of the vibrational frequency issues identified in our prior work [17]. DFT results in Figure 4c and Table

To explore the limit of the best performing PMF-CFG models, we further increased the number of integration time steps for inference and show the results for $\Delta\epsilon$, ϵ_{HOMO} and ϵ_{LUMO} . Increasing the time steps slightly reduce the property MAEs for these three properties, but it has negligible results for the other three properties (Table

Generated Molecules Represent High Structural Validity and Diversity

To examine the chemical validity and diversity of generated molecules, we evaluate the molecules using molecule stability, RDKit validity and uniqueness ratio. Molecule stability evaluates whether the bonding of each atom satisfies simple chemical valency–electron rules. For example, a zero-charge N atom should have a valency (bond order) of 3 where an N atom with +1 charge should carry a valency of 4. In addition, a molecule is only stable if the total net charge is zero. RDKit validity examines whether molecules adhere to perception rules so as to ensure consistency of molecular structures within the RDKit framework [27]. A molecule is unique if its SMILES is different from others. Results for molecule stability indicate that all guidance methods outperform GeoLDM and GCDM, which omit bond orders, and edge out the previous SOTA, JODO, with explicit bond information in molecular graph representations (Figure 5a). Although PMG-CFG achieves the best property alignment, it incurs a 2–3.4% decline in molecule stability compared to the Vanilla conditional model for $\Delta\epsilon$, ϵ_{LUMO} , μ , and C_v ; results for α and ϵ_{HOMO} remains essentially unchanged. In contrast, the PMF-AG models improve molecule stability across all properties relative to PMF-Vanilla, likely because amplifying the differences between main and guide models steer samples away from poorly modeled regions and towards well-learned ones. Despite their weaker property alignment, MG models surpass their vanilla counterparts in molecule stability, achieving the highest molecule stability for $\Delta\epsilon$ and μ . The lower molecule stability of CFG, compared to AG and MG, can also be attributed to the higher discrete guidance weights of CFG against those of AG and MG (Table

To quantify diversity, we calculated the proportion of generated molecules that are both RDKit valid and unique in their SMILES representation. The ‘Valid and Unique’ rates for each method are shown in Figure 5b, and per-property results are summarized in Table

Having examined these individual performance aspects, we now present a holistic comparison to better understand the fundamental trade-offs between guidance methods. We compared all guidance methods across four dimensions—property alignment, structural validity, uniqueness, and sampling efficiency (Figure 5d). For visualization clarity, each metric has been min-max scaled to the range of [0, 1] (see Section

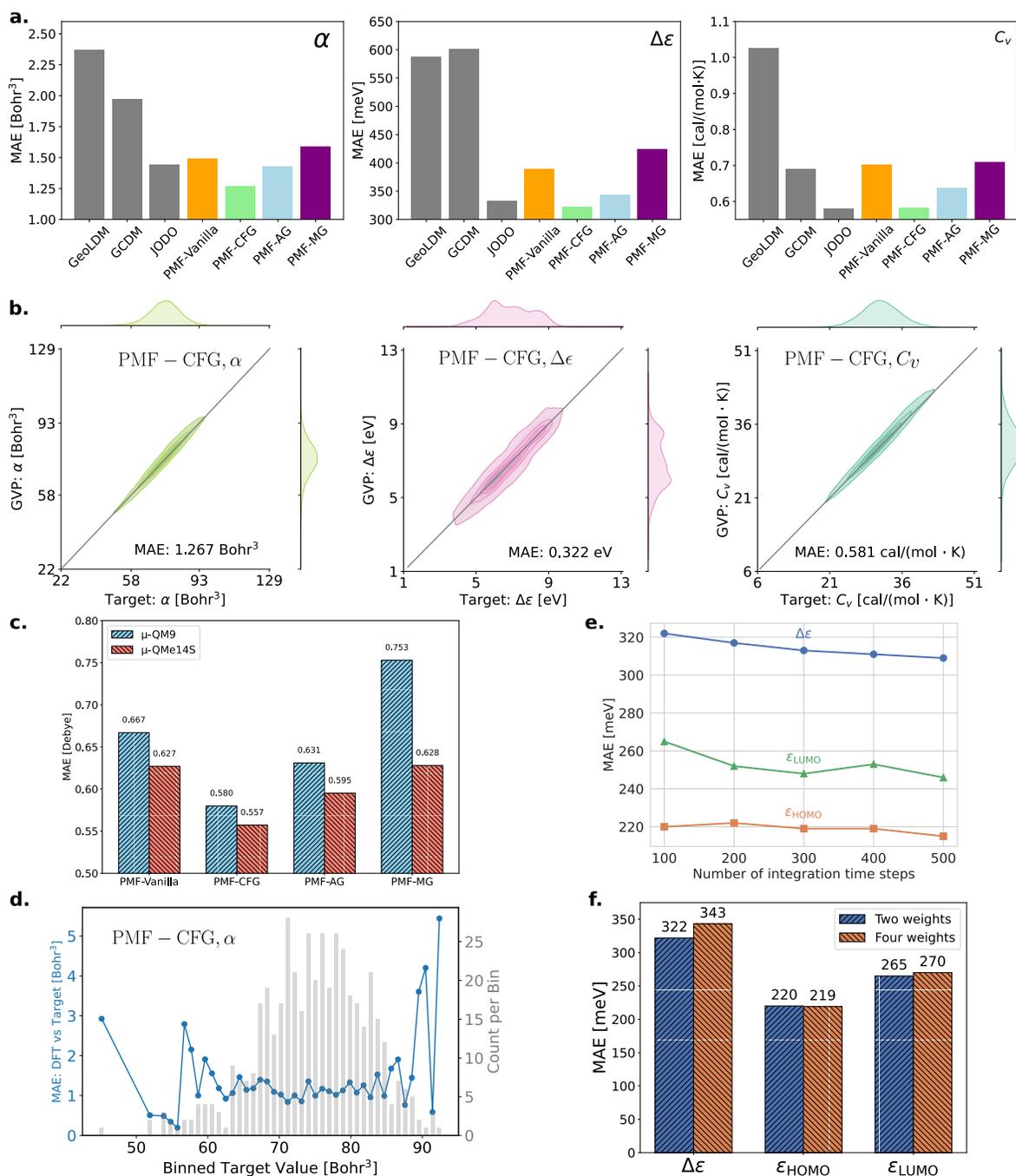


Figure 4: Property alignment performance using different guidance methods. a. Comparison between our guidance methods with three baseline models GeoLDM, GCDM and JODO using property MAEs. Lower numbers indicate better performance. b. One-to-one comparison between target and GVP-predicted property values. c. Performance comparison for models trained on QM9 and QMe145 datasets over dipole moment (μ). d. Mean absolute difference between DFT and Target property values (Left axis) and the target value distribution (Right axis). e. Ablation study on the number of integration steps for PMF-CFG conditioned on $\Delta\epsilon$, ϵ_{HOMO} and ϵ_{LUMO} . f. Property MAEs using Bayesian optimized two guidance weights against those using four guidance weights for PMF-CFG conditioned on $\Delta\epsilon$, ϵ_{HOMO} and ϵ_{LUMO} .

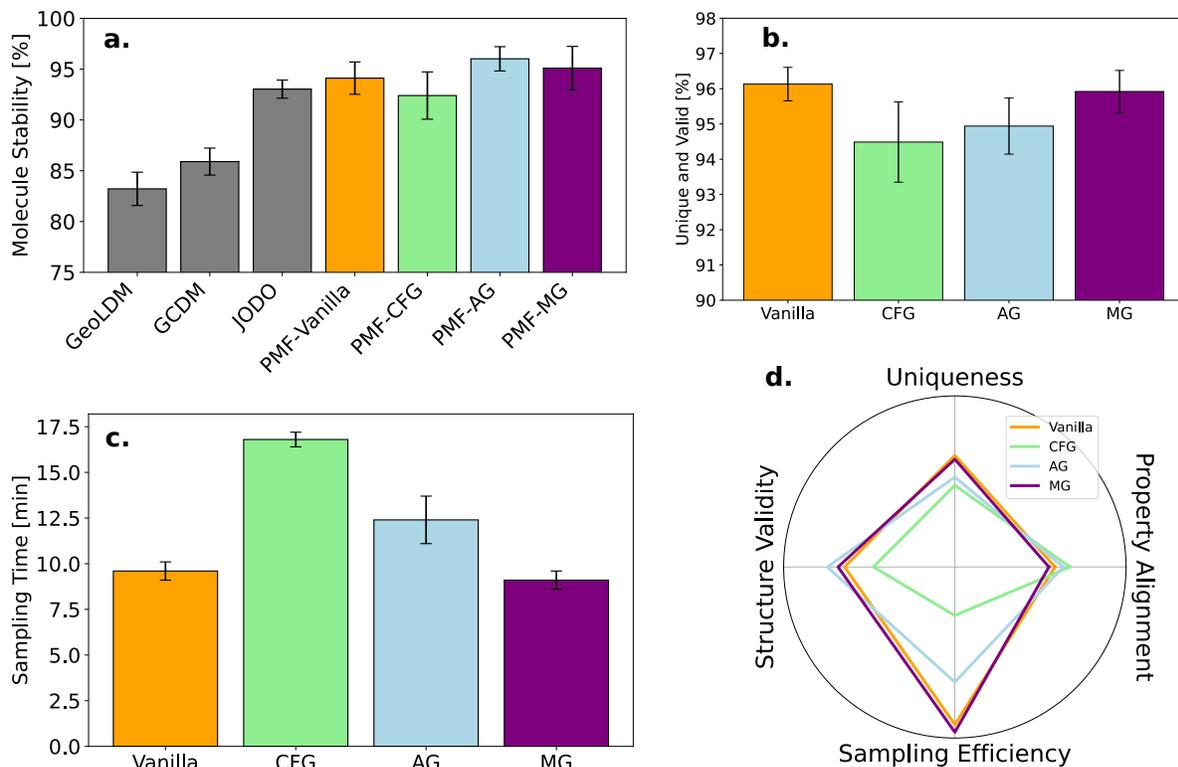


Figure 5: Performance in molecule stability, RDKit validity and uniqueness, sampling efficiency and overall performance for models trained on the rQM9 data. (a) Molecule stability of guidance methods against three baseline conditional models. (b) RDKit validity and uniqueness ratio across our guidance methods. (c) Sampling efficiency across different guidance methods. (d) Overall performance in four dimensions. Results in (a), (b), (c) are reported as the mean \pm std across six molecular properties.

Discussion

This work presents a comprehensive study of advanced guidance methods for conditional molecule generation. Our framework demonstrates robust transferability from QM9 (5 elements) to QMe14S (14 elements) and pushes a new state-of-the-art performance for property alignment and structural validity.

While our work lays a solid foundation for conditional molecule generation, current study is still focused on a single scalar property. Extending the methods to high-dimensional properties such as tensorial properties and to multi-property joint conditions require methodological development and future investigation. Also, although we studied two strategies to construct the guide model in AG, further exploring the guide models, such as EMA strategy [12], may give rise to better performance. Besides, guidance based on the linear interpolation of velocity fields may break on general Riemannian folds [28] or with a non-Gaussian base distribution and this line of research represents a promising future direction. Moreover, recent efforts also focus on inference time optimization and add a control support to steer the generation towards desired directions during sampling [29, 30].

Despite these limitations and challenges, the diverse guidance methods introduced and guidance format discussed here are readily transferrable to other scientific domains involving multi-modality representations, including *de novo* generation for both inorganic and organic materials and protein design, thereby broadening the impact of robust conditional generation across chemistry, materials science, and beyond.

Methods

Molecular Representation

Molecules are represented by fully-connected graphs G . Graph nodes encode the atomic types A^i , charges C^i and positions X^i of a molecule,¹ and edges are the bond orders between two atoms E^{ij} which are found to enhance structure validity and stability for generated molecules [31, 20, 32]. Therefore, a molecule can be denoted as $G = (X, A, C, E)$, where $X = \{X^i\}_{i=1}^N$, $A = \{A^i\}_{i=1}^N$, $C = \{C^i\}_{i=1}^N$ and $E = \{E^{ij} | i \neq j, i, j \in \{1, 2, \dots, N\}\}$ are the atomic positions, atom types, charges and bond orders, respectively. Note that atom types, charges and bond orders are discrete categorical variables, while atomic positions are continuous variables.

Joint Flow Matching for Molecule Generation

Each molecular modality and the molecular graph is learned through a joint flow matching process parameterized by an SE(3) equivariant graph neural network. Equivariance is critical to improve the model expressivity to describe equivariant properties, as molecules are geometric objects whose properties, such as atomic forces or hyper-polarizability, are subject to equivariance [33, 34, 35, 36]. The SE(3) equivariant framework implemented in FlowMol [37, 20] is used as opposed to E(3) frameworks [6, 7] because breaking the reflection symmetry can generate different molecules due to chirality [38, 39]. PropMolFlow used here for vanilla conditional generation, is the property-conditioned implementation of FlowMol. [17]. A linear interpolant is used for all molecular modalities. Vanilla flow matching is adopted for continuous variables, such as atomic positions, whereas discrete modalities, like atom types, bond order, and charges are evolved by discrete flow matching using a Continuous Time Markov Chain (CTMC) process, proposed by [21] and [40]. Interaction between each molecular modality is achieved by a sequence of node feature, node position, and edge feature updates. Interpolant and loss function details can be found in Supporting Information

Vanilla Conditional Generation

Conditional generation allows users to generate samples aligned with specific requirements. This is typically achieved with generative processes parameterized by neural networks that take the target property as an input. Specifically, we control the outcome by choosing a property and generating a sample from the conditional distribution $p(x_t|c)$ where c is the condition, for example, the property, label or text prompt, and x_t is the noisy data. In practice, this can be achieved by training a denoiser network $\epsilon_\theta(x_t, t, c)$ for diffusion models or a conditional velocity field $u_\theta(x_t, t, c)$ for a flow matching generative process. Taking flow matching as an example, the learning objective can be written as:

$$\mathcal{L}(\theta) = \mathbb{E}_{t, p_{t|1}(x_t|z, c), p_z} [\|u_\theta(x_t, t, c) - u_t\|] \quad (6)$$

where u_t is the target conditional velocity field at time t , and z is a conditioning variable that is normally chosen as $z = (x_0, x_1)$ representing both the initial and final states from a base and target distribution, respectively. This type of conditional generation is often termed as *Vanilla* conditional generation without explicit guidance.

Classifier-free Guidance

In computer vision, vanilla conditional generation trained on complex visual datasets often struggle to reproduce training images due to the finite capacity of neural nets [12]. To improve sample quality, *classifier guidance* was introduced by [8]. This approach employs an auxiliary classifier $p_\theta(c|x_t)$ to perform low-temperature sampling by amplifying data points for which the classifier assigns a high likelihood to the target label. It approximates a modified distribution:

$$\tilde{p}_\theta(x_t|c) = p_\theta(x_t|c) \cdot p_\theta(c|x_t)^{w-1} \quad (7)$$

Assigning $w > 1$ serves as a guidance scale that steers the sampling process toward regions of high classifier confidence. While effective at increasing alignment with the desired class, classifier guidance requires training an additional classifier on noisy intermediate data, and relies on classifier’s gradient, $\nabla_{x_t} \log p_\theta(y|x_t)$, to direct samples toward high-likelihood regions, often at the expense of sample diversity.

Classifier-free guidance (CFG) is an alternative to classifier guidance with the same effect but does not rely on gradients from a classifier [9]. In a CFG approach for flow matching, we train a velocity field $u_\theta(x_t, t, \emptyset)$ without property conditioning and a velocity field with property conditioning $u_\theta(x_t, t, c)$. During training, a portion of property labels—typically $p_{\text{uncond}} = 0.1$ [9, 13]—are dropped and replaced with empty labels to allow both conditional and unconditional objectives to be learned within a single framework. During sampling, CFG requires two forward

¹We denote the atom index using a superscript; e.g., position of atom i is denoted X^i

passes—one pass with conditioning and another without conditioning—to generate samples, thereby nearly doubling the computational overhead at sampling compared to a vanilla conditional model. The inference linearly interpolates between these two velocity fields with a weight w :

$$\hat{u}_\theta(x_t, t, c; w) = (1 - w) \cdot u_\theta(x_t, t, \emptyset) + w \cdot u_\theta(x_t, t, c) \quad (8)$$

where w is the guidance weight controlling the strength of conditioning. $w = 0$ recovers unconditional generation, and $w = 1$ corresponds to vanilla conditional generation. This linear interpolation of velocity fields is equivalent to linearly interpolating the score functions—the gradients of the log probability density—when the prior is a standard Gaussian [41], which holds for our *de novo* molecule generation setting and results in the modified distribution in Eq. 7. Values of $w > 1$ are often used to further amplify the conditioning signal, which typically leads to stronger adherence to the condition c , but also potentially at the cost of even lower sample diversity and validity. This reduced validity and diversity have been attributed to the failure of the unconditional model, $u_\theta(x_t, t, \emptyset)$, which faces a more difficult task compared to the conditional model because it only takes a small portion of training budget given by p_{uncond} while attempting to generate all classes at once [12].

Autoguidance

Autoguidance (AG), introduced by [12], uses a high-quality main model D_m along with a poor guide model D_g trained on the same task, conditioning, and data splits, but D_g is intentionally degraded, for instance, by having lower model capacity or shorter training. This approach is termed *Autoguidance* because it uses a bad version of itself to guide the generation. The guide model D_g is expected to make similar errors in the same regions as the main model D_m , and by subtracting the predictions of the guide model from that of the main model and amplifying the differences by a guidance weight, it pushes the generation away from the weaker model and toward better samples. In a flow-matching setting, if we denote the main and guide model as $u_m(x_t, t, c)$ and $u_g(x_t, t, c)$, respectively, the interpolated velocity field at a given weight w reads as:

$$\hat{u}(x_t, t, c; w) = w \cdot u_m(x_t, t, c) + (1 - w) \cdot u_g(x_t, t, c) \quad (9)$$

In practice, the main and guide models should be carefully selected to ensure an appropriate quality gap. The two models should carry similar degradations to remain compatible, while ensuring that the differences are large enough to outweigh random effects such as random initialization of neural networks and random shuffling of training data [42].

Model Guidance

Model Guidance (MG) [13] offers an alternative to CFG by directly modifying the training objective to include an implicit guidance signal. Instead of training two separate conditional and unconditional models, MG plugs the guidance weight into the model’s training target. The model guidance loss then becomes:

$$\mathcal{L}_{\text{MG}}(\theta) = \mathbb{E}_{t, p_{t|1}(x_t|z, c), p_z} [\|u_\theta(x_t, t, c) - \hat{u}_t\|] \quad (10)$$

$$\hat{u}_t = u_t + w \cdot \text{sg}(\hat{u}_\theta(x_t, t, c) - \hat{u}_\theta(x_t, t, \emptyset)) \quad (11)$$

where u_t is the ground-truth velocity field and \hat{u}_t is the modified target. A stopping gradient operation (‘sg’) is applied to avoid model collapse [43]. An Exponential Moving Average (EMA) counterpart of the online velocity field, $\hat{u}_\theta(\cdot)$, is normally used to smooth training and provide more stable model predictions. In addition, the guidance scale/weight w can be fed into neural networks as an additional conditioning input. The model then learns different guidance weights which offer sampling flexibility to balance sample quality and sample diversity, and it has high sampling efficiency since only one forward for sampling is needed. How guidance weights are designed for different proportions of data is provided in the Supporting Information. However, it also creates a difficulty for the models to assimilate the complex interaction between guidance weight embeddings, property embeddings, and molecular graphs.

Table 1: Comparison of Guidance Approaches

Method	Extra Model?	Sampling Cost	Flexible Weights?
Vanilla Conditional Generation	No	1 forward	N/A
Classifier-Free Guidance	No [†]	2 (cond + uncond)	Yes
Autoguidance	Guide model	2 (main + guide)	Yes
Model Guidance	No	1 forward	Optional

[†]Unconditional pass uses same network.

To summarize, we present the key features of each guidance method in Table 1. All methods are implemented within a flow matching generative framework for molecular generation. While most existing techniques are designed to guide

continuous-state generative processes, recent advances have extended guidance to discrete-state spaces [11], exploring guidance schemes without special training procedures [44], and investigated how dimensionality impacts the guidance effects [45].

Discrete guidance: Relationship between denoising probability and instantaneous rate matrix entry

The work by [11] proves that in the setting of classifier guidance (or predictor guidance) for a CTMC process. As show in Figure 2, two key processes require clarification. The first is the conversion from denoising probability to a rate matrix, and the second is the conversion from the rate matrix to transition probabilities.

Denoising Probability to Rate Matrix. In discrete flow matching with masking processes [21], the neural network learns a denoising model $p_\theta(x_1|x_t)$ that predicts the final, clean state x_1 given the current noisy state x_t at time t . This denoising probability will convert to a rate matrix R_t to enable sampling via the CTMC framework.

The rate matrix $R_t^\theta(x_t, \cdot)$ used during sampling is obtained as an expectation over the learned denoising distribution [21]:

$$R_t^\theta(x_t, \cdot) = \mathbb{E}_{p_{1|t}^\theta(x_1|x_t)} [\bar{R}_t(x_t, \cdot|x_1)] \quad (12)$$

where $\bar{R}_t(x_t, \cdot|x_1)$ is the data-conditional rate matrix induced by the forward (noising) process $p_{t|1}(x_t|x_1)$.

For the masking process specifically, where $p_{t|1}(x_t|x_1) = \text{Categorical}(t\delta_{x_1, x_t} + (1-t)\delta_{M, x_t})$, and δ is the Kronecker delta function. The rate matrix has two components [21]: a unmasking entry and a remasking entry.

The rate matrix entry for demasking—from mask token M to actual states—can be written as:

$$R_{t, \text{unmask}}^\theta(x_t, \tilde{x}) = (1 - \delta_{\tilde{x}, M}) \cdot p_\theta(x_1 = \tilde{x}|x_t) \cdot \frac{1 + \eta t}{1 - t} \cdot \delta_{x_t, M} \quad (13)$$

The rate matrix for remasking reads as:

$$R_{t, \text{remask}}^\theta(x_t, M) = \eta \cdot (1 - \delta_{x_t, M}) \quad (14)$$

where η is the stochasticity parameter that controls the probability of remasking. The total rate matrix is then:

$$R_t^\theta(x_t, \tilde{x}) = R_{t, \text{unmask}}^\theta(x_t, \tilde{x}) + R_{t, \text{remask}}^\theta(x_t, M) \quad (15)$$

The diagonal elements $R_t(x_t, x_t)$ are set to ensure conservation of probability flow:

$$R_t^\theta(x_t, x_t) = - \sum_{\tilde{x} \neq x_t} R_t^\theta(x_t, \tilde{x}) \quad (16)$$

This conversion allows the discrete denoising probability output by the neural network to define the instantaneous transition rates needed for CTMC-based sampling [21].

Rate Matrix to Transition Probability. The probability of jumping from a state x at time t to the next state \tilde{x} at time $t + \Delta t$ is given by:

$$p(x_{t+\Delta t} = \tilde{x}|x_t = x, y) = \delta_{x, \tilde{x}} + \frac{p(y|\tilde{x}, t)}{p(y|x, t)} \cdot R_t(x, \tilde{x}) \cdot \Delta t + \mathcal{O}(\Delta t^{1+\epsilon}) \quad (17)$$

where Δt defines an infinitesimal time step in the continuous time space. y is the desired property to contion on. $p(y|x_t)$ indicates a predictor/classifier that relates a noisy state sampled at time t to the property, which can be obtained by minimizing a cross-entropy loss. $\delta_{x, \tilde{x}}$ is the Kronecker function, which can be also denoted as $\delta(x, \tilde{x})$. $R_t(x, \tilde{x})$ indicates the unconditional rate matrix, or it can be written as $R_t(x, \tilde{x}|\emptyset)$. $\mathcal{O}(\Delta t^{1+\epsilon})$ is used to denote terms that decay to zero faster when $t \rightarrow 0$.

Introducing an inverse guidance temperature $w = 1/T$, which is normally termed *guidance strength*, Eq. (17) can be converted to:

$$p^{(w)}(x_{t+\Delta t} = \tilde{x}|x_t = x, y) = \delta_{x, \tilde{x}} + \left[\frac{p(y|\tilde{x}, t)}{p(y|x, t)} \right]^w \cdot R_t(x, \tilde{x}) \cdot \Delta t + \mathcal{O}(\Delta t^{1+\epsilon}) \quad (18)$$

$w > 1$ corresponds to a low-temperature sampling, pushing the samples more toward the conditional generation. Then we can also write:

$$R_t^{(w)}(x, \tilde{x}|y) = \left[\frac{p(y|\tilde{x}, t)}{p(y|x, t)} \right]^w \cdot R_t(x, \tilde{x}) \quad (19)$$

Reformulating Eq. (19) using Bayes’s theorem, and simplifying it with Taylor expansion (see the Supporting Information in [11]), one can obtain the corresponding classifier-free guidance on rate matrix:

$$R_t^{(w)}(x, \tilde{x}|y) = R_t(x, \tilde{x}|y)^w \cdot R_t(x, \tilde{x}|\emptyset)^{1-w} \quad (20)$$

Using a linear interpolant, the rate matrix in the formulation of [21] can be also written as:

$$R_t(x, \tilde{x}) = \frac{1 + \eta t}{1 - t} \cdot p_{1|t}^\theta(x_1 = \tilde{x}|x_t = x) \cdot \delta(x, M) + \eta \cdot (1 - \delta(x, M)) \cdot \delta(\tilde{x}, M) \quad (21)$$

Eq. (21) holds for both conditional and unconditional generation. If $\eta = 0$, the above equation can be reduced:

$$R_t(x, \tilde{x}) = \frac{p_{1|t}^\theta(x_1 = \tilde{x}|x_t = x)}{1 - t} \cdot \delta(x, M) \propto p_{1|t}^\theta \quad (22)$$

Hence it can be readily shown:

$$R_t^{(w)}(x, \tilde{x}|y) = \frac{p_{1|t}^{(w)}(x_1 = \tilde{x}|x_t = x, y)}{1 - t} \cdot \delta(x, M) \quad (23)$$

where $R_t^{(w)}(x, \tilde{x}|y)$ is defined in Eq. (20) and $p_{1|t}^{(w)}(x_1 = \tilde{x}|x_t = x, y)$ refers to the definition in Eq. (3) for guidance on discrete variables.

Evaluation metrics

We evaluate guidance methods along four complementary axes. Property alignment examines how closely the generated molecules’ properties match the target properties used during sampling. This is evaluated by MAEs of property values of generated molecules compared to the input target values. For molecule sampling, each molecule’s atom count (n) is first drawn from the distribution of QM9 and QMe14S training data, and its property values (c) is then conditioned on n to respect the joint distribution on $p(n, c)$.

For structural validity/stability, molecule stability checks the charge–valency consistency. Molecular stability is given by proportions of molecules that all atoms are stable and the the net charge of the molecules are zero if charges are included in the molecular graphs. An atom is stable if it has the correct valency given the formal charge it carries. For instance, a C atom is stable if it has a valency of 4 without charge but a valency of 3 with a -1 charge. In addition, we also report model performance on RDKit sanitization validity [27] and PoseBusters validity [46]. RDKit validity corresponds to proportions of molecules that pass the RDKit sanitization. PoseBusters validity indicates proportions of molecules that pass *de novo* chemical and structural validity tests using the first ten columns of PoseBusters outputs, including sanitization, all atoms connected, valid bond lengths, valid bond angles, no internal steric clashes, and flat aromatic rings. Structural diversity is evaluated by the ‘valid and unique’ ratios, bond-order entropy, element entropy and scaffold diversity. ‘Valid and unique’ ratios quantify the number of molecules that can be sanitized in RDKit and are unique in their SMILES representation. Bond-order entropy is the base-2 Shannon entropy measure of how diverse the bond-type distribution is across a set of generated molecules. Here, four types of bonds—single, double, triple and aromatic bonds—are considered. We first counter the global total of each bond type n_i across generated molecules, and use the counts to generate probability distribution p_i where $p_i = n_i / \sum_i n_i$. The Shannon entropy is then given by: $H = -\sum_i p_i \log_2(p_i)$. Element entropy is computed analogously to bond-order entropy but measures the diversity of atom type distributions across the generated molecules. For each element type (e.g., C, H, O, N, F for QM9; additionally B, Al, Si, P, S, Cl, As, Se, Br for QMe14S), we count the global total of each element n_i across all generated molecules and compute the Shannon entropy using the same formula. Higher element entropy indicates more diverse elemental compositions. Scaffold diversity quantifies the structural variety of molecular frameworks using Murcko scaffolds [47], which represent the core ring systems and linkers of molecules while removing side chains. We generate the Murcko scaffold for each molecule and compute scaffold diversity as the ratio of unique scaffold structures to the total number of generated molecules. Higher scaffold diversity indicates that the model generates molecules with more varied core structures rather than repeatedly generating molecules based on the same scaffolds.

Computational efficiency includes training time and sampling efficiency. Training time is evaluated using the same hardware settings (see more details in

Scaling factors for metrics in the radar plot. Below we elaborate the definition of each metric in the radar plot (Figure 5). If a higher value is preferred (*e.g.*, Bond Diversity, Structure Validity), the metric is transformed via the Eq. 24 (left); otherwise the Right equation is used.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \text{ and } x' = \frac{x_{\max} - x}{x_{\max} - x_{\min}} \quad (24)$$

Sampling efficiency uses the min and max scaling factors of 8 and 20 minutes, respectively. Structure validity is the average of molecule stability and RDKit validity, and the min and max scaling factors are 90% and 100%, respectively. Uniqueness uses the same scaling min and max as that of structure validity. Property alignment is quantified by the MAEs between the GVP-predicted property values for generated molecules and the input target property values. Lower values are better. The min and max scaling factors are the QM9 lower bound and the QM9 bound given by the # Atoms shown in Table

Dataset

We evaluate our methods on two molecular datasets of increasing complexity. The first dataset is the QM9 SDF dataset, which provides explicit bond orders and atomic charges [48]. The original SDF data was found to carry a significant number of charge and bond errors; so instead we use the corrected SDF dataset ‘rQM9_v0.sdf’ for training. [17, 49]. After RDKit sanitization, 133k molecules remained and were further partitioned into 100k training, 20k validation and 13k test samples. The 100k training set was then split evenly following previous work [6]: one subset of 50k for training the PropMolFlow molecular generation models and a second subset of 50k for training the property predictor. Details of the original QM9 data are provided in Supporting Information

To assess the scalability of our guidance methods on more diverse chemical spaces, we introduced the QMe14S dataset [18], which contains 186k small organic molecules spanning 14 elements (H, B, C, N, O, F, Al, Si, P, S, Cl, As, Se, and Br). Since this dataset lacks explicit bond and charge information, we employed Open Babel [50] to infer these molecular attributes. Following RDKit sanitization, 183k molecules remained and were partitioned into 140k training, 25k validation, and 18k test samples. Following the same protocol as QM9, we allocated 70k data for the PropMolFlow model and 70k for training property predictors.

Implementation Detail

Property-Conditioned Molecule Generation

Property-conditioned generation is achieved by the interaction between the node features of a randomly sampled noisy molecular graph and a property embedding. The property embedding is generated by projecting scalar molecular properties (*e.g.*, dipole moment μ) to a high-dimensional latent space through a shallow multi-layer perceptron (MLP). Specifically, for AG and CFG, we employ a ‘concatenate_sum’ operation: the property embedding is concatenated to the node features, then projected to a latent space via a MLP to match the dimension of node features, followed by a summation operation. In prior work we have shown that this choice of operation works well across all properties [17]. Following the original MG work [13], a ‘sum’ operation is used for MG. Starting from the property-conditioned node feature, the molecular graph is iteratively updated through the joint flow matching process to generate the final molecule.

Guidance Implementation

We implemented and compared four variants of guidance in the PropMolFlow framework: vanilla conditional generation, classifier-free guidance, autoguidance, and model guidance.

CFG uses $p_{\text{uncond}} = 0.1$, which denotes the probability of training on unconditional generation during joint training of the conditional and unconditional flow matching models. During sampling, we applied separate guidance weights for continuous atomic positions and discrete variables (atom types, charges and bond orders) to accommodate the hybrid joint flow matching.

AG uses two types of guide models, $u_g(x_t, t, c)$, with reduced training time and/or decreased model complexity. For under-training, we saved model checkpoints every 20,000 training steps (around every 51 epochs with a batch size of 128). For model parameter reduction, we decreased both the hidden dimension size and the number of message-passing layers in the GNN vector field model, reducing the parameter count from 7.1M to 313K.

For MG, we incorporated the guidance weight as an additional input condition, allowing the model to adapt to flexible guidance weights during sampling. We maintained $p_{\text{uncond}} = 0.1$ for which guidance weights are set as zeros, and set the proportion of training examples with model guidance to 0.2 and their guidance weights are randomly chosen between 1 and 2, and the remaining data is treated as the vanilla conditional model for which the guidance weights

are ones. The model guidance is introduced after 10,000 training steps. To obtain the modified target velocity field in Eq. (10), EMA uses a decay rate of 0.9999.

Bayesian Optimization of Guidance Weights

The MAEs between target properties and properties of generated molecules are modeled as the optimization objective. This Bayesian optimization was performed on top of a Gaussian process via the Scikit-Optimize library [51]. We initialized the search with 10 randomly sampled guidance weights to bootstrap the surrogate model, followed by 40 acquisition-function evaluations using the expected improvement (EI) criterion. To ensure a relatively high structural validity, for AG, w_1 and w_2 are optimized in the range of [1, 4.3] and [1, 1.8], respectively, whereas for CFG both weights are optimized in the range of [1, 4]. The Bayesian optimization for AG is performed on two guide models, and the guide model with the lowest MAE for each property is reported hereafter. A scale-aware MG is utilized and the same guidance weights are employed for both atomic positions and discrete variables. Bayesian optimization for the single MG guidance weight starts with 5 initial points, followed by 10 function evaluations. For each guidance weight candidate, 1000 molecules were sampled and the objective MAEs were calculated on these molecules. This procedure was repeated independently for each molecular property.

Training details and hyperparameters

All guidance models on the QM9 and QMe14S datasets were trained with 8 molecule update blocks. Atoms contain 256 hidden scalar features and 16 hidden vector features. Edges contain 128 hidden features. These models were trained with 2000 epochs with a learning rate of 0.00025 together with an Adam optimizer [52] is used for learning the neural networks. Training and inference for PropMolFlow models used a single NVIDIA A100-SXM4 graphic card with 80GB memory with a batch size of 128. All models can be trained in about 2–4 days.

GVP regressor details

To be self-consistent, we trained property regressors using Graph Neural Networks based on GVPs. We appended an MLP layer that takes the final node scalar features as read-out layers to predict the target property. The parameters of GVP regressors are optimized by minimizing a mean squared error loss. Separate models were trained for each of the six tested properties. For the QM9 dataset, GVP training uses a disjoint 50k split compared to the other 50k split used for training the PropMolFlow generative model. For the QMe14S dataset, 70k molecules are used for training the generative model and the other 70k for training the regressor for the property μ .

Data availability

The SDF files containing the molecular structures for the QM9 and QMe14S datasets are available at <https://zenodo.org/records/15700961> and <https://zenodo.org/records/16847162>.

Code availability

The code to reproduce this work is available at <https://github.com/Liu-Group-UF/MolGuidance>.

Acknowledgment

The authors are grateful to Philipp Höllmer for bringing the model guidance paper to our attention at a journal club meeting. This work was supported by NSF Grant OAC-2311632 and the AI and Complex Computational Research Award of the University of Florida. S.M acknowledges support from the Simons Center for Computational Physical Chemistry (Simons Foundation grant 839534). The author thank UFIT Research Computing, and the NVIDIA AI Technology Center at UF for computational resources and consultation.

References

- [1] Yuanqi Du, Arian R Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Liò, Philippe Schwaller, and Tom L Blundell. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, 6(6):589–604, 2024.

- [2] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018.
- [3] Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27(8):675–679, 2013.
- [4] Olivier J Wouters, Martin McKee, and Jeroen Luyten. Estimated research and development investment needed to bring a new medicine to market, 2009-2018. *Jama*, 323(9):844–853, 2020.
- [5] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.*, 4(2):268–276, February 2018.
- [6] Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant Diffusion for Molecule Generation in 3D. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8867–8887. PMLR, June 2022.
- [7] Minkai Xu, Alexander S Powers, Ron O. Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3D molecule generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38592–38610. PMLR, 23–29 Jul 2023.
- [8] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- [9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [10] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, pages 1–3, January 2025.
- [11] Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [12] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [13] Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025.
- [14] Yanchen Luo, Zhiyuan Liu, Yi Zhao, Sihang Li, Hengxing Cai, Kenji Kawaguchi, Tat-Seng Chua, Yang Zhang, and Xiang Wang. Towards unified and lossless latent space for 3d molecular latent diffusion modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [15] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.
- [16] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.
- [17] Cheng Zeng, Jirui Jin, George Karypis, Mark Transtrum, Ellad B. Tadmor, Richard G. Hennig, Adrian Roitberg, Stefano Martiniani, and Mingjie Liu. Propmolflow: Property-guided molecule generation with geometry-complete flow matching. *arXiv preprint arXiv:2505.21469*, 2025.
- [18] Mingzhi Yuan, Zihan Zou, Yi Luo, Jun Jiang, and Wei Hu. Qme14s: A comprehensive and efficient spectral data set for small organic molecules. *The Journal of Physical Chemistry Letters*, 16(16):3972–3979, 2025.
- [19] Majdi Hassan, Nikhil Shenoy, Jungyoon Lee, Hannes Stärk, Stephan Thaler, and Dominique Beaini. Et-flow: Equivariant flow-matching for molecular conformer generation. *Advances in Neural Information Processing Systems*, 37:128798–128824, 2024.
- [20] Ian Dunn and David R. Koes. Exploring discrete flow matching for 3d de novo molecule generation. *arXiv preprint arXiv:2411.16644*, 2024.

- [21] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Forty-first International Conference on Machine Learning*, 2024.
- [22] Cheng Zeng, Jirui Jin, and Mingjie Liu. Propmolflow data: Version v3. Zenodo. <https://doi.org/10.5281/zenodo.17726328>, 2025.
- [23] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.
- [24] Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3D molecule generation and optimization. *Commun Chem*, 7(1):1–11, July 2024.
- [25] Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. Learning Joint 2-D and 3-D Graph Diffusion Models for Complete Molecule Generation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11857–11871, September 2024.
- [26] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. Gaussian-16 Revision C.01, 2016. Gaussian Inc. Wallingford CT.
- [27] Greg Landrum et al. Rdkit: open-source cheminformatics software. RDKit, 2016.
- [28] Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.
- [29] Jaihoon Kim, TaeHoon Yoon, Jisung Hwang, and Minhyuk Sung. Inference-time scaling for flow models via stochastic generation and rollover budget forcing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [30] Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky T. Q. Chen. Adjoint schrödinger bridge sampler. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [31] Clément Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In Danaï Koutra, Claudia Plant, Manuel Gomez Rodriguez, Elena Baralis, and Francesco Bonchi, editors, *Machine Learning and Knowledge Discovery in Databases: Research Track*, pages 560–576, Cham, 2023. Springer Nature Switzerland.
- [32] Tuan Le, Julian Cremer, Frank Noe, Djork-Arné Clevert, and Kristof T Schütt. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. In *The Twelfth International Conference on Learning Representations*, 2024.
- [33] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9323–9332. PMLR, 18–24 Jul 2021.
- [34] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat Commun*, 13(1):2453, May 2022.
- [35] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds, May 2018.
- [36] Yuzhi Xu, Daqian Bian, Cheng-Wei Ju, Fanyu Zhao, Pujun Xie, Yuanqing Wang, Wei Hu, Zhenrong Sun, John Z. H. Zhang, and Tong Zhu. Pretrained E(3)-equivariant message-passing neural networks with multi-level representations for organic molecule spectra prediction. *npj Computational Materials*, 11(1):203, July 2025.
- [37] Ian Dunn and David Ryan Koes. Mixed continuous and categorical flow matching for 3d de novo molecule generation. *CoRR*, abs/2404.19739, 2024.
- [38] Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L. Blundell, Pietro Lio, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909, December 2024. Publisher: Nature Publishing Group.

- [39] Alexandru Dumitrescu, Dani Korpela, Markus Heinonen, Yogesh Verma, Valerii Iakovlev, Vikas Garg, and Harri Lähdesmäki. E(3)-equivariant models cannot learn chirality: Field-based molecular generation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [40] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [41] Carles Domingo-Enrich, Michal Drozdal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [42] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. *arXiv preprint arXiv:2312.02696*, 2024.
- [43] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284, 2020.
- [44] Seyedmorteza Sadat, Manuel Kansy, Otmar Hilliges, and Romann M. Weber. No training, no problem: Re-thinking classifier-free guidance for diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [45] Krunoslav Lehman Pavasovic, Jakob Verbeek, Giulio Biroli, and Marc Mezard. Classifier-free guidance: From high-dimensional analysis to generalized guidance forms. *arXiv preprint arXiv:2502.07849*, 2025.
- [46] Martin Buttenschoen, Garrett M. Morris, and Charlotte M. Deane. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.
- [47] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.
- [48] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- [49] Cheng Zeng, Jirui Jin, and Mingjie Liu. Propmolflow data: Version v3. Zenodo. <https://doi.org/10.5281/zenodo.17726328>, 2025.
- [50] Noel M O’Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):33, 2011.
- [51] Tim Head, Gilles Louppe, Iaroslav Shcherbatyi, and Christopher Schröder. scikit-optimize/scikit-optimize: v0.5.2. Zenodo, March 2018. Software package.
- [52] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.