MolGuidance: A Comparative Study of Guidance Methods for Conditional Molecule Generation

Cheng Zeng [*1] **Jirui Jin** [*1] **Pawan Prakash** [1] **George Karypis** [2] **Mark Transtrum** [3] **Ellad B. Tadmor** [2]
Richard G. Hennig [1] **Adrian Roitberg** [1] **Stefano Martiniani** [4] **Mingjie Liu** [1]

Abstract

Key objectives in conditional molecular generation include ensuring chemical validity, aligning generated molecules with target properties, and promoting diversity and novelty. Recent advances in computer vision introduce a range of new guidance strategies that can be adapted for 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 probabilities, respectively—while jointly optimizing their guidance scales via Bayesian optimization. Our implementation, benchmarked on the QM9 dataset, achieves a 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.

1. Introduction

The generation of novel molecular structures with desired chemical and biological properties is crucial for drug design. Deep learning-based generative models have shown immense promise in accelerating the quality and rate of materials and chemical discovery (Du et al., 2024; Sanchez-Lengeling & Aspuru-Guzik, 2018; Gómez-Bombarelli et al., 2018; Zeng et al., 2024; Zeni et al., 2025). 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, such as therapeutic efficacy, synthetic accessibility, or desired quantum chemical attributes (Du et al., 2024; Gao & Coley, 2020).

Various guidance mechanisms have been proposed to steer generative models. Vanilla conditional generation takes the property/condition and molecule representation (*e.g.*, a molecular graph) as a joint input, and it learns a shared representation for properties and molecular structures that drives the sampling (Hoogeboom et al., 2022; Xu et al., 2023). While the vanilla conditional generation proves to be simple and effective, more advanced guidance techniques have been proposed to improve the generation quality in multiple dimensions. Seminal guidance methods include classifier guidance (Dhariwal & Nichol, 2021), to classifier-free methods (Ho & Salimans, 2022; Zeni et al., 2025), and more recent autoguidance (Karras et al., 2024a) and model guidance (Tang et al., 2025) that 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. In a first kind of implementation, we adopted advanced property guidance methods in the context of *de novo* molecular generation using flow matching. We aim to evaluate and benchmark four methods for conditional generation — including the vanilla conditional generation, classifier-free guidance, autoguidance, and model guidance — 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 pros and cons of each guidance method and provide insights into the current landscape of guidance techniques for molecule generation. Our specific contributions are summarized below:

[*]Equal contribution [1]University of Florida [2]University of Minnesota [3]Brigham Young University [4]New York University. Correspondence to: Mingjie Liu <mingjieliu@ufl.edu>, Stefano Martiniani <sm7683@nyu.edu>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. We present the first implementation of different guidance methods for *de novo* conditional molecule generation in 3D using an SE(3) flow matching process.

2. We introduced a hybrid guidance strategy that separately handles continuous and discrete molecular modalities within classifier-free guidance and autoguidance frameworks.

3. To achieve optimal property alignment, we employed Bayesian optimization to jointly tune their guidance weights.

4. Our guidance methods achieved the new state-of-the-art performance for property alignment and structure validity metrics.

2. Background

2.1. 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, *e.g.*, 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:

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

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.

2.2. 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 (Karras et al., 2024a). To improve sample quality, *classifier guidance* was introduced by (Dhariwal & Nichol, 2021). 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:

$$\tilde{p}_\theta(x_t|c) = p_\theta(x_t|c) \cdot p_\theta(c|x_t)^{w-1} \quad (2)$$

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 (Ho & Salimans, 2022). In a CFG approach for flow matching, we train a velocity field $u_\theta(x_t, t, \varnothing)$ without property conditioning and a velocity field with property conditioning $u_\theta(x_t, t, c)$. During training, a portion of property labels—typically $p_{\text{uncond}} = 0.1$ (Ho & Salimans, 2022; Tang et al., 2025)—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 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:

$$\hat{u}_\theta(x_t, t, c; w) = (1-w)\cdot u_\theta(x_t, t, \varnothing) + w \cdot u_\theta(x_t, t, c) \quad (3)$$

where w is the guidance weight controlling the strength of conditioning. $w = 0$ recovers unconditional generation, and $w = 1$ corresponds to vanilla conditional generation. 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, $u_\theta(x_t, t, \varnothing)$, 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 (Karras et al., 2024a).

2.3. Autoguidance

Autoguidance (AG), introduced by (Karras et al., 2024a), 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:

$$\hat{u}(x_t, t, c; w) = w \cdot u_m(x_t, t, c) + (1 - w) \cdot u_g(x_t, t, c) \quad (4)$$

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 (Karras et al., 2024b).

2.4. Model Guidance

Model Guidance (MG) (Tang et al., 2025) 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:

$$\mathcal{L}_{\text{MG}}(\theta) = \mathbb{E}_{t, p_{t|1}(x_t|z,c), p_z} \left[\| u_\theta(x_t, t, c) - \hat{u}_t \| \right] \quad (5)$$

$$\hat{u}_t = u_t + w \cdot \text{sg}\left(\hat{u}_\theta(x_t, t, c) - \hat{u}_\theta(x_t, t, \emptyset) \right) \quad (6)$$

where u_t is the ground-truth velocity filed and $\hat{u}_t$ is the modified target. A stopping gradient operation ('sg') is applied to avoid model collapse (Grill et al., 2020). 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 subsequent Section 3.4. 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 (Nisonoff et al., 2025), exploring guidance schemes without special training procedures (Sadat et al., 2025), and investigated how dimensionality impacts the guidance effects (Pavasovic et al., 2025).

2.5. Flow Matching

Flow matching is the state-of-the-art generative process that learns a velocity field connecting a base distribution to a target distribution (Liu et al., 2022; Albergo et al., 2023; Lipman et al., 2023; Gagneux et al., 2025). By conditioning on samples from the base and target distributions, the conditional velocity field parameterized by a neural network aims to reconstruct the probability path bridging the two distributions. The generation process proceeds by first sampling a noisy state from the base distributions and then propagating the state forward in time by solving a neural ODE (Chen et al., 2019). Flow matching offers multiple levels of design flexibility, such as base distributions, probability paths (viz., interpolants) and ODE solvers. It has proved to be more robust than diffusion models in various applications, from image generation (Lipman et al., 2023; Ma et al., 2024) to crystal materials generation (Miller et al., 2024; Hoellmer et al., 2025), molecule generation (Song et al., 2023; Dunn & Koes, 2024b; Zeng et al., 2025b) and protein structure prediction (Campbell et al., 2024).

3. Method

Our approach integrates a hybrid guidance approach with a flow matching framework for conditional molecular generation in 3D (Figure 1). For the vanilla property-conditioned generation we adopt PropMolFlow (Zeng et al., 2025b). This flow matching framework work utilizes a concatenation and sum operation to describe the interaction between a property embedding and node features (Figure 1(b)). Since molecules consist of both continuous and discrete modalities, a hybrid guidance approach is employed (Figure 1(c)), and guidance weights are optimized via Bayesian optimization targeting property mean absolute errors (MAEs); Figure 1(d).

3.1. 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 (Vignac et al., 2023; Dunn & Koes, 2024b; Le et al., 2023). Therefore, a molecule can be denoted as $G = (X, A, C, E)$,

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

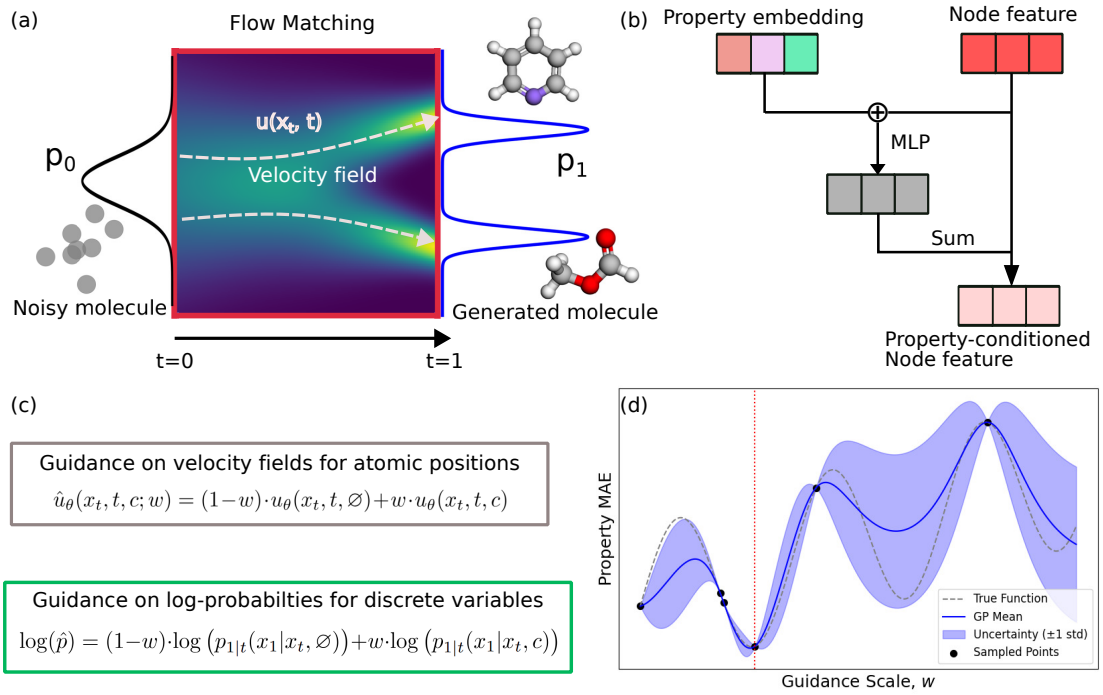

Figure 1: Conditional molecular generation with flow matching and Bayesian optimized guidance scales. (a) Molecular generation is achieved through a flow matching generative process by learning a velocity field. (b) Property conditioning is achieved by a concatenation operation followed by a summation. (c) A hybrid guidance scheme where guidance on atomic positions and discrete variables (e.g., atom types and bond orders) are imposed on the velocity fields and predicted log-probability, respectively. Here CFG is used as the example. (d) A schematic for Bayesian optimization of guidance scale to minimize property MAEs.

where $X = \{X^i\}_{i=1}^N$, $A = \{A^i\}_{i=1}^N$, $C = \{C^i\}_{i=1}^N$ and $E = \{E^{ij}|i \neq j, i, j \in \{1, 2, \cdots, N\}\}$ 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.

3.2. 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 (Satorras et al., 2022; Batzner et al., 2022; Thomas et al., 2018; Xu et al., 2025). The SE(3) equivariant framework implemented in FlowMol (Dunn & Koes, 2024a;b) is used as opposed to E(3) frameworks (Hoogeboom et al., 2022; Xu et al., 2023) because breaking the reflection symmetry can generate different molecules due to chirality (Jing et al., 2021; Schneuing et al., 2024). PropMolFlow used here for vanilla conditional generation, is the property-conditioned implementation of FlowMol. (Zeng et al., 2025b). 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 (Campbell et al., 2024) and (Gat et al., 2024). 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 Appendix A, and Model details can be found in Appendix B.

3.3. 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 (Zeng et al., 2025b). Following the original MG work (Tang et al., 2025), 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.

3.4. 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 $p_{\text{uncondo}} = 0.1$, 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. Following Eq. (2), we implemented guidance in the logarithmic domain for discrete variables as shown in Figure 1(c). We show in Appendix Section C that when there is no stochasticity in the CTMC process, our approach is equivalent to the guidance on the rate matrix as proved in (Nisonoff et al., 2025).

AG uses two types of guide models, $u_{\text{g}}(x_t, t, c)$, 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. (5), EMA uses a decay rate of 0.9999.

3.5. Bayesian Optimization of Guidance Weights

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 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 (Head et al., 2018). 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.

4. Experiments

In this work, we systematically investigate how different guidance methods affect small-molecule generation on the QM9 dataset (Ramakrishnan et al., 2014; Wu et al., 2018). We evaluate each method's conditional generation performance across four evaluation dimensions: property alignment, structural stability/validity, structural diversity and computational efficiency.

4.1. Setup

Dataset. We used the QM9 SDF dataset, which provides explicit bond orders and atomic charges (Wu et al., 2018). The original SDF data was found to carry a significant number of charge and bond errors; so instead we use the corrected rQM9 SDF dataset (Zeng et al., 2025b;a). 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 (Hoogeboom et al., 2022): one subset of 50k for training the PropMolFlow molecular generation models and a second subset of 50k for training the property predictor. Conditional generation was evaluated for six property labels, including polarizability (α), HOMO-LUMO gap, HOMO energy, LUMO energy, dipole moment (μ), and heat capacity (C_v). Details of the rQM9 SDF data and computational settings for model training are provided in Appendix G. The widely used large-size GEOM-Drug conformer dataset (Axelrod & Gómez-Bombarelli, 2022) was not considered in this study, as it does not provide the quantum mechanical labels required for our analysis.

Baseline Models. We evaluate the conditional generation results against three diffusion-based baseline models: GCDM (Morehead & Cheng, 2024), GeoLDM (Xu et al.,

2023) and JODO (Huang et al., 2024). For consistency, we report only each baseline's conditional generation performance, and their unconditional results are available in the original publications. GCDM and GeoLDM directly operate on 3D point clouds without explicit bond orders, so we assign bond orders post hoc using optimized cutoff distances from (Hoogeboom et al., 2022). In contrast, JODO inherently models both bond orders and aromaticity. None of the baselines' conditional generation incorporate atomic charges. We compare all guidance methods with baselines on structural validity/stability and property alignment. In addition, we compare different guidance strategies on structural diversity and computational efficiency.

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. For structural validity/stability, molecule stability checks the charge–valency consistency. In addition, we also report model performance on PoseBusters validity (Buttenschoen et al., 2024) and RDKit sanitization validity (RDKit, 2024). Structural diversity is quantified in terms of ratios of molecules that are RDKit valid and unique which is also denoted as 'uniqueness' in following discussions, as well as bond-order entropy. Computational efficiency accounts for the training duration and sampling efficiency. During sampling, PropMolFlow integrates the learned/interpolated velocity fields via Euler's method over 100 evenly spaced time steps. Each molecule's atom count (n) is first drawn from the distribution of QM9 training data, and its property values (c) is then conditioned on n to respect the joint distribution on $p(n, c)$. Precise definitions of all evaluation metrics are included in Appendix E.

4.2. Model performance

Bayesian Optimized Guidance Weights. Figure 2 illustrates Bayesian optimization of CFG weights for the HOMO property. Across the search space, the MAEs vary narrowly between around 200 and 280 meV, with far greater sensitivity to the discrete-modality weight (w_2) than to the atomic-position weight (w_1). Optimal performance (MAE=202 meV) occurs at w $(w_1, w_2) = (2.71, 1.91)$, and lower MAEs tend to cluster near $w_2 = 2$.

Another example for the AG model conditioned on C_v (Figure 5, Appendix F), reveals a similar dependence on the two guidance weights. Table 9 in Appendix lists the best weights for CFG, AG and MG. Table 10 in Appendix shows the variation of property MAEs for Bayesian optimization. It is crucial to identify the optimal guidances as the MAEs can vary by more than ten times in certain situations (e.g., from 1.40 to 15.5 Bohr^3 for AG conditioned on α). The guide model that yields the best performance under AG is provided in Appendix 11. As a general rule, greater differences

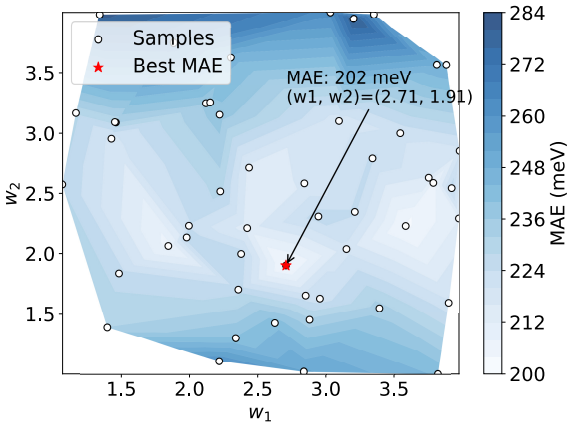

Figure 2: Performance of Bayesian optimization over guidance weights for CFG conditioning on HOMO energy. Hollow circles denote sampled weight pairs, with the best candidate (lowest MAE) highlighted by a red star; its guidance weights and MAE are indicated. MAEs were computed over 1000 molecules sampled from the joint distribution $p(n, c)$.

between the guide and main models tend to produce lower property MAEs. Notably, many optimal atomic-position weights (w_1) lie at the edges of the search range, suggesting that expanding this range may offer minimal additional improvement given its weaker MAE dependence. An example for the MG model is provided in Appendix Figure 6. All subsequent performance metrics are computed using models with the optimal guidance weights.

Property Alignment. We perform a quantitative comparison of conditional generation across different guidance approaches and the three baseline models GeoLDM, GCDM and JODO. Results are shown in Table 2. The "Random" corresponds to MAEs between original molecular properties and fully shuffled properties, removing any correlations between structures and properties, hence serving as an upper bound. The "# Atoms" baseline uses atom counts as the predictor for molecular properties. The "QM9" baseline uses a separate predictor trained on a disjoint 50k molecules to predict properties of the 50k molecules used to train the generative model. The property predictor is trained on the QM9 xyz data (Ramakrishnan et al., 2014), and provided by Hoogeboom *et al.* (Hoogeboom et al., 2022), and the corresponding MAE serves as a lower bound on achievable error. Improvement over the "# Atoms" baseline suggests that the generative model captures structural features beyond simple atom count when generating new molecules. All PMF guidance methods outperform the baseline models without bond order (GeoLDM, GCDM) by a large margin. In particular, PMF-CFG achieves the best alignment for four properties—α, $\Delta\epsilon$, ϵ_{HOMO} and μ—while remaining competitive on ϵ_{LUMO} and $C_v$against the state-of-the-art JODO model. PMF-AG surpasses PMF-Vanilla across all properties, and

Table 2: Mean Absolute Error for molecular property prediction (lower is better). PropMolFlow (PMF) results employ Bayesian-optimized guidance weights. Top-ranked values are **bold**, second-best values are underlined. JODO results are from our own sampled molecules.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Units	Bohr^3	meV	meV	meV	Debye	cal/(mol·K)
QM9 (Lower-Bound)	0.10	64	39	36	0.043	0.040
Random (Upper-Bound)	9.01	1470	645	1457	1.616	6.857
# Atoms	3.86	866	426	813	1.053	1.971
GeoLDM	2.37	587	340	522	1.108	1.025
GCDM	1.97	602	344	479	0.844	0.689
JODO	1.44	333	231	**260**	0.620	**0.580**
PMF-Vanilla	1.49	390	266	325	0.667	0.702
PMF-CFG	**1.27**	**322**	**220**	265	**0.580**	0.581
PMF-AG	1.43	344	242	274	0.631	0.638
PMF-MG	1.59	425	273	346	0.753	0.708

it outperforms JODO on α, matches its performance on $\Delta\epsilon$, ϵ_{HOMO}, ϵ_{LUMO} and μ, and it falls slightly behind on C_v. The scale-aware PMF-MG models underperform their vanilla counterparts across all properties, suggesting that jointly learning property constraints and guidance scale embeddings remains challenging for conditional molecule generation.

Since the property predictor shares the same model architecture with the PropMolFlow generative model, it may exhibit an inductive bias that favors good performance on the kind of molecules generated by PropMolFlow. To further confirm the quality of generated molecules using CFG, we performed DFT calculations for 500 molecules selected from the 10k molecules generated at the same level of theory (B3LYP/6-31G(2df,p)) as the QM9 training data using Gaussian (Frisch et al., 2016). 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 described in (Zeng et al., 2025b). DFT results in Table 3 confirms the property alignment of generated molecules, despite a significant underestimation of the DFT MAEs against input target property values for $\Delta\epsilon$, ϵ_{HOMO} and ϵ_{LUMO}.

Table 3: Performance of property alignment for CFG, evaluated using both DFT and a property predictor. Metrics are computed on 500 molecules selected from the 10k generated molecules reported in Table 2, using the same property units. The 500 molecules were furthered filtered by molecule stability, closed-shell valence electron configuration, RDKit validity, PoseBusters validity, and DFT convergence.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
DFT vs Target	1.18	341	261	242	0.586	0.563
GVP vs Target	1.21	293	213	225	0.580	0.528

Molecule Stability. We also evaluate molecule stability against three baselines for each guidance method. PMF-Vanilla not only outperforms by a wide margin GeoLDM and GCDM, which omit bond orders, but also edges out the previous SOTA, JODO with bond orders. Although PMG-CFG delivers the best property alignment, it incurs a 2–3.4 % decline in stability compared to Vanilla for $\Delta\epsilon$, ϵ_{LUMO}, μ, and C_v; results for α and ϵ_{HOMO} remains essentially unchanged. In contrast, the AG models boost 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. RDKit validity (Table 12) and PoseBusters validity (Table 13) results in Appendix F follow similar trends among AG, CFG and MG, although the gains against the vanilla conditional generation become less significant.

Table 4: Performance of molecule stability (%). Higher numbers indicate better performance. All results for baseline models are based on our own sampling. The best results are in **bold** and the second best results are underlined.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
GeoLDM	81.4	83.1	84.0	84.0	85.5	81.3
GCDM	85.0	86.0	88.3	84.7	86.3	85.1
JODO	92.7	94.1	93.5	92.5	93.7	91.7
PMF						
Vanilla	92.8	94.6	95.1	94.2	96.2	91.8
CFG	93.1	92.5	95.6	92.0	92.8	88.4
AG	**95.8**	95.7	**97.2**	**96.9**	96.6	**93.9**
MG	94.8	**96.9**	96.9	93.3	**96.8**	91.9

Structural Diversity. To quantify diversity, we calculated the proportion of generated molecules that are both RDKit valid and unique in their SMILES representation. The results are summarized in Table 5.

Since SMILES has a one-to-one correspondence with 2D molecular graphs, uniqueness serves as a proxy for the quality of guidance on discrete variables. Compared to the vanilla model, CFG exhibits the most notable decline in uniqueness on average, followed by AG and MG, probably because CFG favors higher discrete guidance weights than

Table 5: Ratios of unique RDKit valid molecules ('Uniqueness') across different guidance methods. All values are reported in unit of '%', and higher numbers indicate higher structural diversity. The highest values are in **bold**, and the second highest values are underlined.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Vanilla	**96.0**	**96.6**	96.5	**95.6**	**96.5**	**95.6**
CFG	94.3	95.3	95.4	94.5	95.0	92.3
AG	95.7	93.5	95.6	94.8	94.7	95.2
MG	95.9	96.2	**96.6**	95.0	96.4	**95.6**

AG and MG (Table 9 in Appendix). We also assessed the bond-order entropy of generated molecules and the results are provided in Appendix Table 14.

Training and Sampling Efficiency. Table 6 reports the training and sampling wall-clock time for each guidance method. Because AG uses the PMF-vanilla's model as its main model, its total training time is unchanged. In practice, one needs to either save a checkpoint model as the guide model or to train a lightweight guide model for AG, which only takes a negligible additional 2 hours. CFG slightly reduces training time by skipping the property-embedding MLP operations on 10% of the data for the unconditional model. MG incurs a modest training overhead relative to PMF-Vanilla because it maintains an EMA copy of the online model. The MG model using EMA requires two forwards to obtain its unconditional and conditional predictions (Eq. 5) without gradient computation, and is updated every training step.

During sampling, both MG and PMF-Vanilla require only a single forward pass, making them the fastest. CFG and AG perform two forward passes—conditional and unconditional passes for CFG, main's and guide's passes for AG—so they take more time for sampling. AG is faster than CFG in sampling because its guide network is much smaller than CFG's unconditional model.

Table 6: Training and sampling times. Training is for 2000 epochs and sampling is for 10 k molecules. Values are reported as mean $\pm$ standard deviation across the six properties.

Model	Training [h] $\downarrow$	Sampling [min] $\downarrow$
Vanilla	55.4 ± 0.7	9.6 ± 0.5
CFG	54.9 ± 0.5	16.8 ± 0.4
AG	57.4 ± 0.7	12.4 ± 1.3
MG	90.3 ± 3.2	9.1 ± 0.5

4.3. Ablations

Guidance weights. Figure 3 shows how molecule stability varies with guidance weights. For both AG and CFG, increasing the weight on atomic positions (w_1) while decreasing the weight on discrete variables (w_2) enhances molecule stability. However, AG is far more sensitive to changes in w_2 than CFG: AG's molecule stability plummets from over 0.9 at $w_2 = 1$ to about 0.1 at $w_2 = 3$, whereas CFG experiences a much smaller drop (≤ 0.15). This suggests that CFG models are more robust to choices of discrete-variable weights, likely because CFG uses a single network with alternate property embeddings for unconditional and conditional generation, while AG relies on two more loosely coupled networks whose logits of discrete-variable predictions may not be compatible when interpolated with large weights.

The guidance-weight variation for property alignment closely mirror the trends observed in our Bayesian optimization experiments (see Appendix Figure 9). Nevertheless, MG models exhibit almost no dependence of molecule stability, bond entropy and property alignment on guidance weights (Appendix Table 17), suggesting that MG struggles to leverage the guidance scaling effects—a phenomenon that merits further investigation.

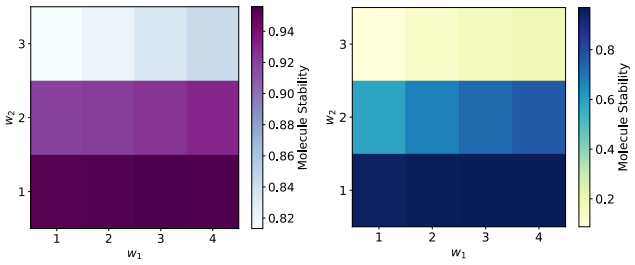

Figure 3: Dependence of molecule stability on guidance weights for CFG (Left) and AG (Right). Values are reported by averaging across six properties.

Inference timesteps for CFG. For a good balance between sampling efficiency and accuracy, we use 100 integration timesteps to generate molecules. Table 7 reports the property MAEs for CFG inference with varying numbers of time steps (n_{ts}). Raising n_{ts} to 200 causes all MAEs to be lower than those of JODO (Table 2). Further increasing n_{ts} improves the alignment for $\Delta\epsilon$, ϵ_{HOMO}, and ϵ_{LUMO}, but has negligible impact on other properties.

Table 7: MAEs for CFG with optimized guidance weights across varying integration timesteps (n_{ts}). Best results are bolded.

n_{ts}	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
100	1.27	322	220	265	0.580	0.581
200	1.24	317	222	252	**0.552**	**0.568**
300	**1.22**	313	219	248	0.559	0.573
400	1.22	311	219	253	0.564	0.575
500	1.23	**309**	**215**	**246**	0.559	0.574

Guidance on all four molecular modalities. The joint flow matching process enables independent control the guidance weight for each individual molecular modality (po-

sitions, atom types, atom charges and bond orders). Table 8 compares the property MAEs for CFG using Bayesian-optimized guidance weights across all four modalities . The results show that using four separate guidance weights yields performance comparable to using just two: one for positions and one shared across all discrete variables. Similar results are observed for AG (Appendix Table 18).

Table 8: Comparison of property MAEs with four guidance weights against two guidance weights for CFG.

CFG	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Two weights	1.27	322	220	265	0.580	0.581
Four weights	1.25	343	219	270	0.571	0.591

5. Discussion and Conclusions

Figure 4 compares all guidance methods across four dimensions—property alignment, structural validity, uniqueness, and sampling efficiency. For visualization clarity, each metric has been min-max scaled to the range of [0, 1] (see Appendix G for definitions and scaling ranges). All three guidance strategies outperform the vanilla model in at least one dimension. The model guidance (MG) approach remains closest to the vanilla model in every dimension, likely because it applies identical weights to both atomic positions and discrete variables, which limits its capability to leverage guidance-weight effects. Classifier-free guidance (CFG) delivers the best property-alignment, outperforming the vanilla models by at least 10 % across six properties, though at the cost of a modest drop in structural validity and uniqueness. Autoguidnace (AG) models also improves property alignment—albeit slightly less than CFG—and boosts structure validity over the vanilla baseline. Both AG and CFG models incur higher computational costs for inference due to two forward passes during sampling.

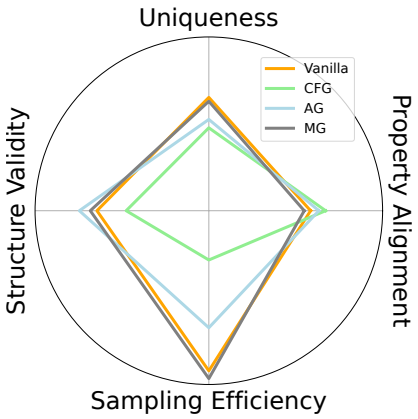

Figure 4: Comparison of guidance methods across four dimensions: property alignment, structure validity, uniqueness and sampling efficiency. For clarity, these metrics are min-max scaled.

In summary, our SE(3) flow-matching framework achieves state-of-the-art results in conditional molecule generation, outperforming prior methods on both structural validity and property alignment. We systematically evaluate three guidance strategies—CFG, AG and scale-aware MG—and highlight their respective strengths and weaknesses against previous models and the vanilla conditional generation. CFG provides the most accurate property matching, whereas AG offers a favorable trade-off between structure validity, uniqueness, and sampling efficiency, although AG models' performance hinges on the guide-model quality and can become unstable at higher guidance weights for discrete variables. While MG has proved to excel in computer vision tasks (Tang et al., 2025), our adaptation to a multi-modal molecular setting has shown that it struggles to incorporate varied guidance scales, underscoring domain-specific challenges.

Looking ahead, extending scale-aware MG to handle distinct guidance weights for different molecular modalities may unlock its full potential. A deeper investigation into the interplay of guidance-scale embeddings, property encoding, and molecular graphs may further bolster MG's performance. For AG, exploring alternative guide models and EMA schedules may improve consistency between the main and guide models. In addition, extending the current approach to a larger dataset, such as PCQM4Mv2 (Hu et al., 2021; Nakata & Shimazaki, 2017), would provide a rigorous evaluation of the scalability and generalizability of current implementations. Finally, the methods introduced here are readily transferrable to other scientific domains, including crystal structure prediction and protein design, thereby broadening the impact of robust conditional generation across chemistry, materials science, and beyond.

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 from 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 authors thank UFIT Research Computing, and the NVIDIA AI Technology Center at UF for computational resources and consultation.

Impact Statement

This paper provides a comprehensive study on property-guided generative models for conditional molecular generation in 3D. The goal is to advance the field of AI for Science and there are many potential downstream applications, none of which we feel necessary to be specified here.

References

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, November 2023. URL http://arxiv.org/abs/2303.08797.

Axelrod, S. and Gómez-Bombarelli, R. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Sci Data*, 9(1): 185, April 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01288-4. URL https://www.nature.com/articles/s41597-022-01288-4.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer Normalization, July 2016. URL http://arxiv.org/abs/1607.06450.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat Commun*, 13(1):2453, May 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29939-5. URL https://www.nature.com/articles/s41467-022-29939-5.

Buttenschoen, M., M. Morris, G., and M. Deane, C. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9): 3130–3139, 2024. doi: 10.1039/D3SC04185A. URL https://pubs.rsc.org/en/content/articlelanding/2024/sc/d3sc04185a.

Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T. Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design, June 2024. URL http://arxiv.org/abs/2402.04997.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural Ordinary Differential Equations, December 2019. URL http://arxiv.org/abs/1806.07366.

Dhariwal, P. and Nichol, A. Diffusion Models Beat GANs on Image Synthesis, June 2021. URL http://arxiv.org/abs/2105.05233.

Du, Y., Jamasb, A. R., Guo, J., Fu, T., Harris, C., Wang, Y., Duan, C., Liò, P., Schwaller, P., and Blundell, T. L. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, 6(6):589–604, 2024.

Dunn, I. and Koes, D. R. Mixed Continuous and Categorical Flow Matching for 3D De Novo Molecule Generation, April 2024a. URL http://arxiv.org/abs/2404.19739.

Dunn, I. and Koes, D. R. Exploring Discrete Flow Matching for 3D De Novo Molecule Generation, November 2024b. URL http://arxiv.org/abs/2411.16644.

Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., Scalmani, G., Barone, V., Petersson, G. A., Nakatsuji, H., Li, X., Caricato, M., Marenich, A. V., Bloino, J., Janesko, B. G., Gomperts, R., Mennucci, B., Hratchian, H. P., Ortiz, J. V., Izmaylov, A. F., Sonnenberg, J. L., Williams-Young, D., Ding, F., Lipparini, F., Egidi, F., Goings, J., Peng, B., Petrone, A., Henderson, T., Ranasinghe, D., Zakrzewski, V. G., Gao, J., Rega, N., Zheng, G., Liang, W., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Vreven, T., Throssell, K., Montgomery, Jr., J. A., Peralta, J. E., Ogliaro, F., Bearpark, M. J., Heyd, J. J., Brothers, E. N., Kudin, K. N., Staroverov, V. N., Keith, T. A., Kobayashi, R., Normand, J., Raghavachari, K., Rendell, A. P., Burant, J. C., Iyengar, S. S., Tomasi, J., Cossi, M., Millam, J. M., Klene, M., Adamo, C., Cammi, R., Ochterski, J. W., Martin, R. L., Morokuma, K., Farkas, O., Foresman, J. B., and Fox, D. J. Gaussian~16 Revision C.01, 2016. Gaussian Inc. Wallingford CT.

Gagneux, A., Martin, S. T., Emonet, R., Bertrand, Q., and Massias, M. A Visual Dive into Conditional Flow Matching. February 2025. URL https://openreview.net/forum?id=M6MTUQc4um.

Gao, W. and Coley, C. W. The Synthesizability of Molecules Proposed by Generative Models. *J. Chem. Inf. Model.*, 60 (12):5714–5723, December 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c00174. URL https://doi.org/10.1021/acs.jcim.0c00174.

Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T. Q., Synnaeve, G., Adi, Y., and Lipman, Y. Discrete Flow Matching, November 2024. URL http://arxiv.org/abs/2407.15595.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised Learning, September 2020. URL http://arxiv.org/abs/2006.07733.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.*, 4(2):268–276, February 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00572. URL https://doi.org/10.1021/acscentsci.7b00572.

Head, T., Louppe, G., Shcherbatyi, I., and Schröder, C. scikit-optimize/scikit-optimize: v0.5.2, March 2018. URL https://zenodo.org/records/1207017.

Ho, J. and Salimans, T. Classifier-Free Diffusion Guidance, July 2022. URL http://arxiv.org/abs/2207.12598.

Hoellmer, P., Egg, T., Martirossyan, M. M., Fuemmeler, E., Gupta, A., Shui, Z., Prakash, P., Roitberg, A., Liu, M., Karypis, G., Transtrum, M., Hennig, R. G., Tadmor, E. B., and Martiniani, S. Open Materials Generation with Stochastic Interpolants, February 2025. URL http://arxiv.org/abs/2502.02582.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant Diffusion for Molecule Generation in 3D. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 8867–8887. PMLR, June 2022. URL https://proceedings.mlr.press/v162/hoogeboom22a.html.

Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs, October 2021. URL http://arxiv.org/abs/2103.09430.

Huang, H., Sun, L., Du, B., and Lv, W. 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. ISSN 2162-2388. doi: 10.1109/TNNLS.2024.3416328. URL https://ieeexplore.ieee.org/document/10589299.

Jing, B., Eismann, S., Soni, P. N., and Dror, R. O. Equivariant Graph Neural Networks for 3D Macromolecular Structure, July 2021. URL http://arxiv.org/abs/2106.03843.

Karras, T., Aittala, M., Kynkäänniemi, T., Lehtinen, J., Aila, T., and Laine, S. Guiding a Diffusion Model with a Bad Version of Itself, December 2024a. URL http://arxiv.org/abs/2406.02507.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and Improving the Training Dynamics of Diffusion Models, March 2024b. URL http://arxiv.org/abs/2312.02696.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980.

Klein, L., Krämer, A., and Noé, F. Equivariant flow matching, November 2023. URL http://arxiv.org/abs/2306.15030.

Le, T., Cremer, J., Noé, F., Clevert, D.-A., and Schütt, K. Navigating the Design Space of Equivariant Diffusion-Based Generative Models for De Novo 3D Molecule Generation, November 2023. URL http://arxiv.org/abs/2309.17296.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow Matching for Generative Modeling, February 2023. URL http://arxiv.org/abs/2210.02747.

Liu, X., Gong, C., and Liu, Q. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, September 2022. URL http://arxiv.org/abs/2209.03003.

Ma, N., Goldstein, M., Albergo, M. S., Boffi, N. M., Vanden-Eijnden, E., and Xie, S. SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers, September 2024. URL http://arxiv.org/abs/2401.08740.

Miller, B. K., Chen, R. T. Q., Sriram, A., and Wood, B. M. FlowMM: Generating Materials with Riemannian Flow Matching, June 2024. URL http://arxiv.org/abs/2406.04713.

Morehead, A. and Cheng, J. Geometry-complete diffusion for 3D molecule generation and optimization. *Commun Chem*, 7(1):1–11, July 2024. ISSN 2399-3669. doi: 10.1038/s42004-024-01233-z. URL https://www.nature.com/articles/s42004-024-01233-z.

Nakata, M. and Shimazaki, T. PubChemQC Project: A Large-Scale First-Principles Electronic Structure Database for Data-Driven Chemistry. *Journal of Chemical Information and Modeling*, 57(6):1300–1308, June 2017. ISSN 1549-9596. doi: 10.1021/acs.jcim.7b00083. URL https://doi.org/10.1021/acs.jcim.7b00083.

Nisonoff, H., Xiong, J., Allenspach, S., and Listgarten, J. UNLOCKING GUIDANCE FOR DISCRETE STATE-SPACE DIFFUSION AND FLOW MODELS. 2025.

Pavasovic, K. L., Verbeek, J., Biroli, G., and Mezard, M. Classifier-Free Guidance: From High-Dimensional Analysis to Generalized Guidance Forms, May 2025. URL http://arxiv.org/abs/2502.07849.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

RDKit. RDKit: Open-Source Cheminformatics, 2024. URL https://www.rdkit.org.

Sadat, S., Kansy, M., Hilliges, O., and Weber, R. M. NO TRAINING, NO PROBLEM: RETHINKING CLASSIFIER-FREE GUIDANCE FOR DIFFUSION MODELS. 2025.

Sanchez-Lengeling, B. and Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361 (6400):360–365, July 2018. doi: 10.1126/science.aat2663. URL https://www.science.org/doi/10.1126/science.aat2663.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) Equivariant Graph Neural Networks, February 2022. URL http://arxiv.org/abs/2102.09844.

Schneuing, A., Harris, C., Du, Y., Didi, K., Jamasb, A., Igashov, I., Du, W., Gomes, C., Blundell, T., Lio, P., Welling, M., Bronstein, M., and Correia, B. Structure-based Drug Design with Equivariant Diffusion Models, September 2024. URL http://arxiv.org/abs/2210.13695.

Song, Y., Gong, J., Xu, M., Cao, Z., Lan, Y., Ermon, S., Zhou, H., and Ma, W.-Y. Equivariant Flow Matching with Hybrid Probability Transport for 3D Molecule Generation. *Advances in Neural Information Processing Systems*, 36:549–568, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/01d64478381c33e29ed611f1719f5a37-Abstract-Conference.html.

Tang, Z., Bao, J., Chen, D., and Guo, B. Diffusion Models without Classifier-free Guidance, February 2025. URL http://arxiv.org/abs/2502.12154.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds, May 2018. URL http://arxiv.org/abs/1802.08219.

Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport, March 2024. URL http://arxiv.org/abs/2302.00482.

Vignac, C., Osman, N., Toni, L., and Frossard, P. MiDi: Mixed Graph and 3D Denoising Diffusion for Molecule Generation, June 2023. URL http://arxiv.org/abs/2302.09048.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Xu, M., Powers, A., Dror, R., Ermon, S., and Leskovec, J. Geometric Latent Diffusion Models for 3D Molecule Generation, May 2023. URL http://arxiv.org/abs/2305.01140.

Xu, Y., Bian, D., Ju, C.-W., Zhao, F., Xie, P., Wang, Y., Hu, W., Sun, Z., Zhang, J. Z. H., and Zhu, T. 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. ISSN 2057-3960. doi: 10.1038/s41524-025-01698-z. URL https://www.nature.com/articles/s41524-025-01698-z.

Zeng, C., Khan, Z., and Post, N. L. Data-efficient and Interpretable Inverse Materials Design using a Disentangled Variational Autoencoder, November 2024. URL http://arxiv.org/abs/2409.06740.

Zeng, C., Jin, J., Karypis, G., Transtrum, M., Tadmor, E. B., Hennig, R. G., Roitberg, A., Martiniani, S., and Liu, M. Hugging Face rQM9, 2025a. URL https://huggingface.co/datasets/colabfit/rQM9.

Zeng, C., Jin, J., Karypis, G., Transtrum, M., Tadmor, E. B., Hennig, R. G., Roitberg, A., Martiniani, S., and Liu, M. PropMolFlow: Property-guided Molecule Generation with Geometry-Complete Flow Matching, May 2025b. URL https://arxiv.org/abs/2505.21469v2.

Zeni, C., Pinsler, R., Zügner, D., Fowler, A., Horton, M., Fu, X., Wang, Z., Shysheya, A., Crabbé, J., Ueda, S., Sordillo, R., Sun, L., Smith, J., Nguyen, B., Schulz, H., Lewis, S., Huang, C.-W., Lu, Z., Zhou, Y., Yang, H., Hao, H., Li, J., Yang, C., Li, W., Tomioka, R., and Xie, T. A generative model for inorganic materials design. *Nature*, pp. 1–3, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08628-5. URL https://www.nature.com/articles/s41586-025-08628-5.

A. Interpolants, Priors and Loss Functions

Flow for atomic positions takes a linear interpolant:

$$X_t = (1-t)X_0 + tX_1 \tag{7}$$

where X_0 and X_1 are initial and final states. The base distribution (prior) for atomic positions use a centered standard normal distribution $p_0(X) = \prod_{i=1}^{N} \mathcal{N}(X_0^i|\mathbf{0}, \mathbb{I}_3)$. The optimization of the conditional velocity field for atomic positions can be reparameterized into the optimization of a denoiser network with an mean squared error (MSE) objective:

$$\mathcal{L}_X = \mathbb{E}_{t,p_t(X_t|X_0,X_1),\pi(X_0,X_1)}\left[\|X_{1|t}^{\theta} - X_1\|\right] \tag{8}$$

Where the joint distribution $\pi(X_0, X_1)$ defines the optimal transport coupling between (X_0, X_1). $X_{1|t}^{\theta}$ represents the predicted atomic position given the state at time t. Details of the optimal transport formulation are provided in Appendix D.

Discrete variables such as atom types and charges are modeled through the CTMC flows (Campbell et al., 2024). The prior distribution is the state in which all atoms are in a masked state, and the generation process is essentially a demasking process. We refer the readers to the FlowMol work for more details of the CTMC flows (Dunn & Koes, 2024b). The objective for these discrete variables takes the cross-entropy format:

$$\mathcal{L}_{\text{CE}} = \mathbb{E}_{t,p_{t|1}(x_t|z),p_z}\left[-\log p_{1|t}^{\theta}(x_1^i|x_t)\right] \tag{9}$$

The total loss for the molecule graph is a weighted linear summation of losses for each molecular modality:

$$\mathcal{L} = \eta_X\mathcal{L}_X + \eta_A\mathcal{L}_A + \eta_C\mathcal{L}_C + \eta_E\mathcal{L}_E \tag{10}$$

Empirically, it is preferential to determin atomic positions first, followed by bonds, charges and atom types. In view of this, the loss weights are chosen to be $(\eta_X, \eta_A, \eta_C, \eta_E) = (3.0, 0.4, 1.0, 2.0)$.

B. Model Architecture

In FlowMol's implementation (Dunn & Koes, 2024a), molecule updates are achieved through layers comprising of Geometric Vector Perceptrons (GVPs). Within each GVP, the molecule graph passes through a sequential steps of Node Feature Update (NFU), Node Position Update (NPU) and Edge Feature Update (EFU). Each node i consists of a position $x_i \in \mathbb{R}^3$, scalar features $s_i \in \mathbb{R}^d$, and vector features $v_i \in \mathbb{R}^{c\times 3}$. Non-zero vector features are involved a cross-product vector operation, which is crucial to break the reflection symmetry, making it an SE(3) equivariant architecture. The scalar feature is a concatenation of atom type and charge vectors; that is, $s_i = [a_i \oplus c_i]$ where '$\oplus$' defines a concatenation operation. Each edge feature corresponds to the bond order, and the permutation invariance of bond orders is realized through taking the sum of bond features from $i \to j$ and $j \to i$;that is, $\widehat{e}^{ij} = \text{MLP}(e_{ij} + e_{ji})$.

Node Feature Update. The node feature takes two steps for its update: the first step generates scalar messages $m_{i\to j}^{(s)}$, and vector messages $m_{i\to j}^{(v)}$ by a function ψ_M which corresponds to a sequential two GVPs:

$$m_{i\to j}^{(s)}, m_{i\to j}^{(v)} = \psi_M\left(\left[s_i^{(l)} \oplus e_{ij}^{(l)} \oplus d_{ij}^{(l)}\right], \left[v_i \oplus \frac{x_i^{(l)} - x_j^{(l)}}{d_{ij}^{(l)}}\right]\right) \tag{11}$$

Where $d_{ij}^{(l)}$ is the distance between nodes i and j at the update layer l. To enrich the neighboring environment, the distance d_{ij} is expanded with a radial basis function (RBF) embedding before fed into the next GVPs or MLPs. A message passing procedure is conducted to update node scalar and vector features by aggregation:

$$s_i^{(l+1)}, v_i^{(l+1)} = \text{LN}\left([s_i^{(l)}, v_i^{(l)}] + \psi_N\left(\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\right)\left[m_{j\to i}^{(s)}, m_{j\to i}^{(v)}\right]\right) \tag{12}$$

Where LN stands for s LayerNormalization operation (Ba et al., 2016), ψ_N is a chain of three GVPs.

Node Position Update. Node-wise operations are used on the updated node scalar and vector features to update node position features:

$$x_i^{(l+1)} = x_i^{(l)} + \psi_P\left(s_i^{(l+1)}, v_i^{(l+1)}\right) \tag{13}$$

Where ψ_P is a sequencial three GVPs in which the final output has 1 vector and 0 scalar features.

Edge Feature Update. Edge features are updated by edge-wise operations that takes the updated node scalar features and node distance as the inputs:

$$e_{ij}^{(l+1)} = \text{LayerNorm}\left(e_{ij}^{(l)} + \text{MLP}\left(s_i^{(l+1)}, s_j^{(l+1)}, d_{ij}^{(l+1)}\right)\right) \tag{14}$$

C. Guidance applied to the probability distribution is functionally equivalent to guidance applied to the rate matrix

The work by (Nisonoff et al., 2025) proves that in the setting of classifier guidance (or predictor guidance) for a CTMC process, the probability of jumping from a state x at time t to the next state $\widetilde{x}$ at time $t + \Delta t$ is given by:

$$p(x_{t+\Delta t} = \widetilde{x}|x_t = x, y) = \delta_{x,\widetilde{x}} + \frac{p(y|\widetilde{x}, t)}{p(y|x, t)} \cdot R_t(x, \widetilde{x}) \cdot \Delta t + \mathcal{O}(\Delta t^{1+\epsilon}) \tag{15}$$

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. $\delta_{x,\widetilde{x}}$ is the Kronecker function, which can be also denoted as $\delta(x, \widetilde{x})$. $R_t(x, \widetilde{x})$ indicates the unconditional rate matrix, or it can be written as $R_t(x, \widetilde{x}|\varnothing)$. $\mathcal{O}(\Delta t^{1+\epsilon})$ is used to denote terms that decay to zero faster when $t \to 0$.

Introducing an inverse guidance temperature $w = 1/T$, which is normally termed *guidance strength*, Eq. (15) can be converted to:

$$p^{(w)}(x_{t+\Delta t} = \widetilde{x}|x_t = x, y) = \delta_{x,\widetilde{x}} + \left[\frac{p(y|\widetilde{x}, t)}{p(y|x, t)}\right]^w \cdot R_t(x, \widetilde{x}) \cdot \Delta t + \mathcal{O}(\Delta t^{1+\epsilon}) \tag{16}$$

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

$$R_t^{(w)}(x, \widetilde{x}|y) = \left[\frac{p(y|\widetilde{x}, t)}{p(y|x, t)}\right]^w \cdot R_t(x, \widetilde{x}) \tag{17}$$

Reformulating Eq. (17) using Bayes's theorem, and simplifying it with Taylor expansion (see the Appendix in (Nisonoff et al., 2025)), one can obtain the corresponding classifier-free guidance on rate matrix:

$$R_t^{(w)}(x, \widetilde{x}|y) = R_t(x, \widetilde{x}|y)^w \cdot R_t(x, \widetilde{x}|\varnothing)^{1-w} \tag{18}$$

Using a linear interpolant, the rate matrix in the formulation of (Campbell et al., 2024) can be also written as:

$$R_t(x, \widetilde{x}) = \frac{1 + \eta t}{1 - t} \cdot p_{1|t}^{\theta}(x_1 = \widetilde{x}|x_t = x) \cdot \delta(x, M) + \eta \cdot (1 - \delta(x, M)) \cdot \delta(\widetilde{x}, M) \tag{19}$$

where η is the stochasticity parameter, which allows unmasked states to transit back to an masked state. Eq. (19) holds for both conditional and unconditional generation. If $\eta = 0$, the above equation can be reduced:

$$R_t(x, \widetilde{x}) = \frac{p_{1|t}^{\theta}(x_1 = \widetilde{x}|x_t = x)}{1 - t} \cdot \delta(x, M) \propto p_{1|t}^{\theta} \tag{20}$$

Hence it can be readily shown:

$$R_t^{(w)}(x, \widetilde{x}|y) = \frac{p_{1|t}^{(w)}(x_1 = \widetilde{x}|x_t = x, y)}{1 - t} \cdot \delta(x, M) \tag{21}$$

where $R_t^{(w)}(x, \widetilde{x}|y)$ is defined in Eq. (18) and $p_{1|t}^{(w)}(x_1 = \widetilde{x}|x_t = x, y)$ refers to the definition in Figure 1(c) for guidance on discrete variables.

D. Equivariant Optimal Transport

To smooth the probability path, we used the equivariant optimal transport to align the initially sampled noisy molecules with the target molecules, as used in previous works (Tong et al., 2024; Song et al., 2023; Klein et al., 2023). This is achieved by optimal permutation of atom node indices and rigid body alignment between base molecules and target molecules, for both of which their center-of-masses are removed to respect the translational invariance (Klein et al., 2023; Hoogeboom et al., 2022).

E. Details of Evaluation Metrics.

Property metrics evaluate the alignment of the properties of generated molecules with the input target properties. We sampled jointly the number of atoms and the property values from the QM9 training data. The target properties and the number of atoms are used to generate molecules. The properties of molecules are calculated by property prediction molecules trained on a disjoint set of QM9 data to ensure there is no data leakage between the property prediction models and molecule generation models.

Exact definitions of each metric are summarized below:

- Molecular stability: 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.
- RDKit validity: Proportions of molecules that pass the RDkit sanitization (RDKit, 2024).
- PoseBusters validity: Proportions of molecules that pass *de novo* chemical and structural validity tests, including sanitization, all atoms connected, valid bond lengths, valid bond angles, no internal steric clashes, and flat aromatic rings.
- Uniqueness: Proportions of molecules that are both RDKit valid and unique in their SMILES representation.
- Bond-order entropy: 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, 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:

$$H = -\sum_i p_i \log_2(p_i)$$

- Computational efficiency includes training time and sampling efficiency. Training time is evaluated using the same hardware settings (see more details in G). Sampling efficiency is evaluated by the sampling time for 10k molecules and averaged across conditional models for all six molecule properties.

F. Additional results

F.1. Bayesian optimized guidance weights for AG, MG and CFG.

Table 9: Bayesian optimized guidance weights (w_1, w_2) for each method. For MG, identical weights apply to both positional and discrete variables. All weights are dimensionless.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
PMF-CFG	(4.00, 1.77)	(4.00, 2.16)	(2.71, 1.91)	(3.97, 2.29)	(4.00, 2.31)	(4.00, 2.00)
PMF-AG	(2.34, 1.00)	(4.26, 1.34)	(2.79, 1.11)	(3.15, 1.17)	(4.29, 1.50)	(2.75, 1.14)
PMF-MG	1.34	1.68	1.89	2.23	2.21	2.14

F.2. Bayesian optimization for examples of AG and MG models.

Ranges of MAEs for Bayesian optimization can be found in Table 10.

Table 10: Ranges of MAEs for Bayesian optimization for different guidance methods. Results are evaluated on 1000 sampled molecules. AG uses the guide model trained with 40000 steps and reduced numbers of node and edge features.

Property Units	α Bohr^3	$\Delta\epsilon$ meV	ϵ_{HOMO} meV	ϵ_{LUMO} meV	μ Debye	C_v cal/(mol·K)
CFG, Minimum MAE	1.20	317	202	249	0.549	0.556
CFG, Maximum MAE	2.68	603	282	757	0.746	1.684
AG, Minimum MAE	1.40	328	231	256	0.591	0.616
AG, Maximum MAE	15.5	425	360	466	0.926	1.023
MG, Minimum MAE	1.53	406	261	333	0.714	0.689
MG, Maximum MAE	1.67	457	280	358	0.778	0.759

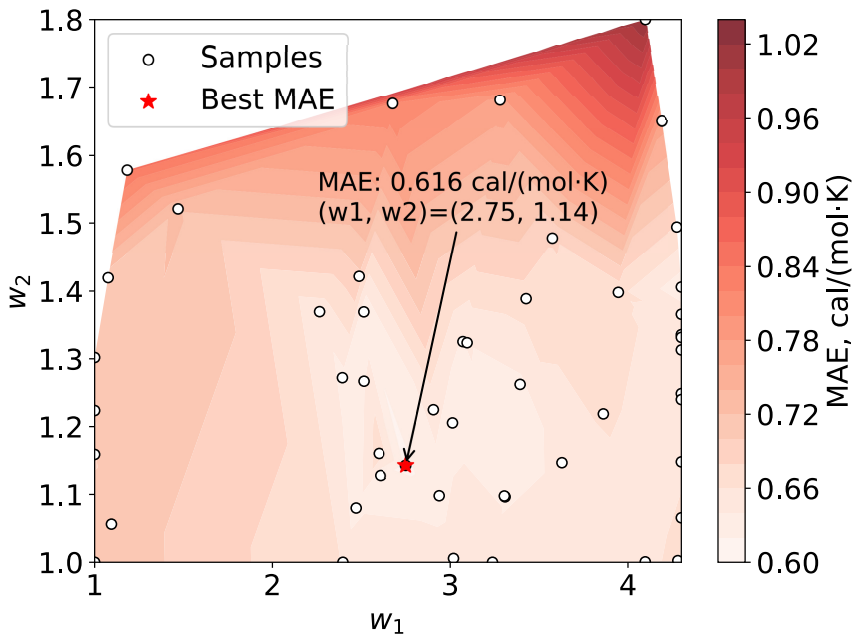

Figure 5: An example of Bayesian optimization for guidance weights for the AG model conditioned in C_v. Samples are shown as hollow circles while the best MAE is marked by a red star and the corresponding weights and MAE are indicated by texts. MAEs are evaluated on 1000 molecules sampled from the joint distribution $p(n, c)$.

Bayesian optimization for the AG model conditioned on C_v is show in Figure 5.

The best guide model for each property is summarized in Table 11. The first guide model ($u_{g,1}$) is trained with 40000 steps and has a reduced-by-half numbers of hidden node features and edge features, and the second guide mode ($u_{g,2}$) is the same architecture of the main model but trained with only 51 epochs.

Table 11: Bayesian optimized guidance weights and the best MAEs for two different guide models in AG. Numbers in bracket correspond to (w_1, w_2), which are dimensionless, and values after the weights are corresponding best MAEs. This evaluation is based on 1000 sampled molecules, and the model chosen for further analysis in the main text are **bold**.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Units	Bohr^3	meV	meV	meV	Debye	cal/(mol·K)
$u_{g,1}$	(2.09, 1.21), 1.40	**(4.26, 1.34), 328**	**(2.79, 1.11), 231**	**(3.15, 1.17), 256**	**(4.29, 1.50), 0.591**	**(2.75, 1.14), 0.616**
$u_{g,2}$	**(2.34, 1.00), 1.39**	(2.47, 1.42), 330	(1.93, 1.28), 234	(3.63, 1.15), 272	(4.29, 1.20), 0.631	(1.00, 1.35), 0.672

Bayesian optimization for the MG model conditioned on $\Delta\epsilon$ is show in Figure 6. MAEs for the MG models vary in a smaller range, suggesting the difficulty in leveraging the guidance weight effects for MG. Also, there are high fluctuations for MAE when guidnace weights changes from around 1.6 to 1.7, indicating the instability of the MG model for describing the guidance weight effect.

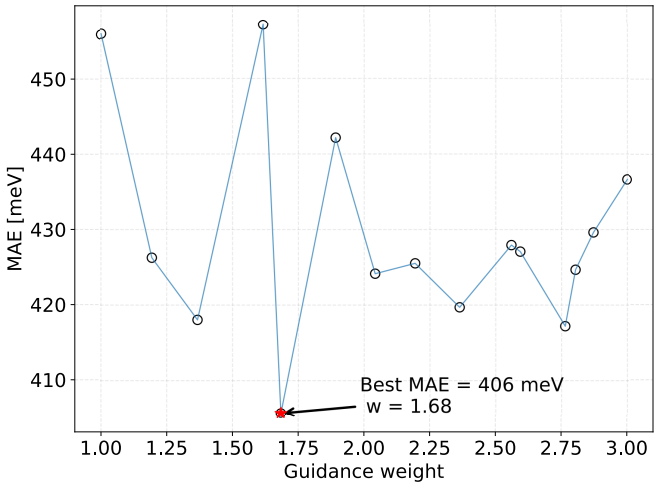

Figure 6: An example of Bayesian optimization for guidance weights for the MG model conditioned in $\Delta\epsilon$. Samples are shown as hollow circles while the best MAE is marked by a red star and the corresponding weights and MAE are indicated by texts. MAEs are evaluated on 1000 molecules sampled from the joint distribution $p(n, c)$.

F.3. Structural Validity

F.3.1. RDKit validity of generated molecules conditioned on six properties

Table 12 shows the RDKit validity of generated molecules conditioning on six properties for baseline models and different guidance methods.

F.3.2. PoseBusters validity of generated molecules conditioned on six properties

Table 13 shows the PoseBuster validity of generated molecules conditioning on six properties for baseline models and different guidance methods.

F.4. Structural Diversity

Table 14 shows that all guidance methods increase bond entropy for most properties compared to the vanilla models. Among guidance approaches, CFG achieves the highest bond entropy for $\Delta\epsilon$, ϵ_{HOMO} and μ, whereas AG has the highest bond entropy for α and C_v. This elevated bond entropy under CFG and AG can likely be attributed to its relatively high guidance

Table 12: RDKit validity (%) for generated molecules conditioned on six properties. All results for baseline models are based on our own sampling and using retrained or publicly available checkpoint models. The best results are in **bond** and the second best results are underlined.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
GeoLDM	91.6	91.8	92.2	92.2	93.0	89.6
GCDM	94.4	94.7	95.4	94.4	94.9	94.4
JODO	96.4	97.0	95.9	95.6	96.4	95.6
PMF						
Vanilla	97.6	**98.4**	98.3	97.8	**98.7**	97.0
CFG	96.8	97.4	97.9	96.6	96.5	94.0
AG	**98.2**	97.3	98.5	**98.0**	98.0	**97.6**
MG	97.6	98.2	**98.9**	96.3	98.6	96.8

Table 13: PoseBuster validity (%) for generated molecules conditioned on six properties. All results for baseline models are based on our own sampling and using retrained or publicly available checkpoint models. The best results are in **bond** and the second best results are underlined.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
GeoLDM	89.1	89.2	90.3	89.9	90.3	87.3
GCDM	91.6	92.0	92.9	92.0	92.4	91.4
JODO	95.2	95.7	94.7	94.5	95.3	94.1
PMF						
Vanilla	95.7	**97.3**	96.6	96.5	97.1	95.5
CFG	95.4	95.3	96.5	94.1	92.9	90.7
AG	**96.7**	94.6	97.3	**97.5**	94.0	**96.2**
MG	95.9	97.1	**97.5**	95.2	**97.5**	95.5

weights on atomic positions (Table 9), which stretch the bond length distributions (Appendix Figure 7, Table 15 and Table 16). By allowing greater variability in bond distances, these settings increase bond entropy.

F.4.1. BOND-ORDER ENTROPY

Table 14: Bond-order entropy of generated molecules under various guidance methods. The highest values are in **bold**, and the second highest values are underlined.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Vanilla	0.593	0.560	0.543	0.495	0.480	0.490
CFG	**0.65**	0.569	0.532	0.562	0.536	**0.617**
AG	0.599	**0.657**	**0.598**	0.556	**0.556**	0.498
MG	0.524	0.570	0.578	**0.567**	0.463	0.510

F.4.2. BOND DISTANCE DISTRIBUTIONS FOR TOP-2 FREQUENT BONDS C-H AND C-C

The bond distance standard deviations for C-H bonds and for C-C bonds can be found in Table 15 and Table 15, respectively.

Table 15: Bond distance standard deviation of generated molecules for each guidance method on the most frequent C-H bonds. All values are in the unit of mÅ. The QM9 C-H bond distance standard deviation is 6.8 mÅ. The highest values are in **bold**, and the second highest values are underlined.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Vanilla	14.2	12.9	11.2	10.9	11.6	12.8
CFG	14.4	16.3	11.7	16.8	17.5	17.2
AG	**28.6**	**57.6**	**32.8**	**34.5**	**52.6**	**26.8**
MG	12.7	14.6	12.1	11.6	9.6	13.8

F.5. Additional Ablations

Analysis of bond entropy (Appendix Figure 8) illustrates that AG consistently produces slightly higher bond entropy than CFG across all weight settings. In both methods, raising either guidance weight increases entropy, with w_2 having a stronger

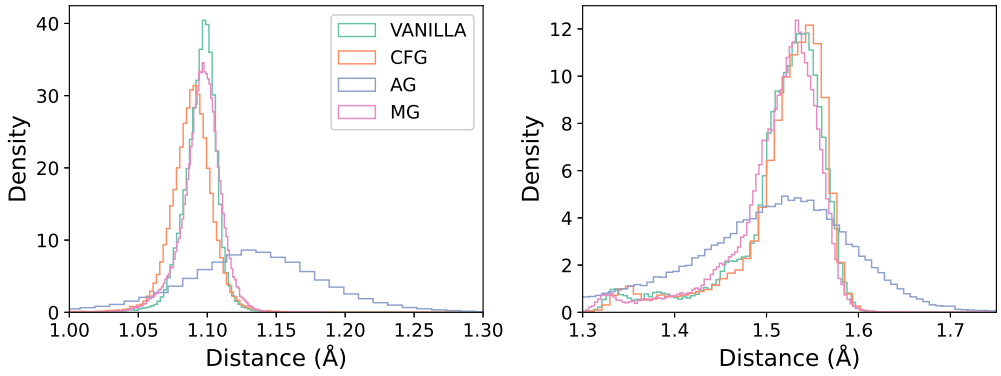

Figure 7: Top-2 frequent bond distance distributions for different guidance methods and the QM9 training data for C-H bonds (Left) and C-C bonds (Right). Molecules are generated by conditioning on the HOMO-LUMO gap ($\Delta\epsilon$).

Table 16: Standard deviation of generated molecules for each guidance method on bond distances of the second most frequent C-C bonds. All values are in the unit of mÅ. The QM9 C-C bond distance standard deviation is 70.8 mÅ. The highest values are in **bold**, and the second highest values are underlined.

Property	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Vanilla	71.8	69.1	65.9	63.7	62.2	68.4
CFG	69.2	75.0	65.9	76.5	74.6	73.4
AG	**86.1**	**106.9**	**86.5**	**86.1**	**93.4**	**82.7**
MG	70.2	70.4	72.3	76.3	59.6	65.3

effect than w_1.

Figure 8 shows the bond entropy *versus* different guidance weights w_1 and w_2.

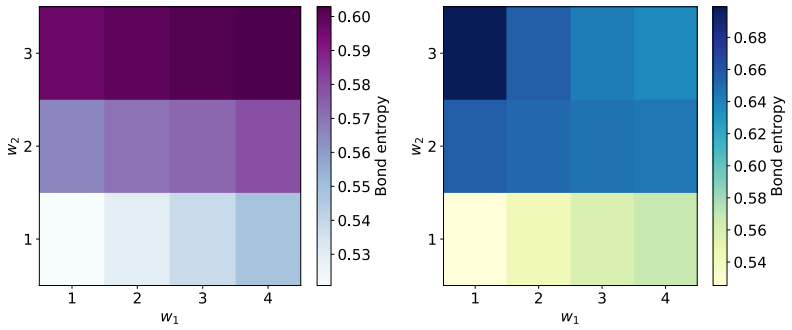

Figure 8: Effects of guidance weights on the bond entropy. Results are averaged across six molecule properties.

Figure 9 shows the property alignment as a function of guidance weights w_1 and w_2. The guidance weights for the lowest MAE of CFG ϵ_{HOMO} are $(w_1, w_2) = (4, 2)$ with a property MAE of 216 meV, and the second lowest MAE comes with $(w_1, w_2) = (3, 2)$ with a MAE of 221 meV, which confirms the findings of the Bayesian analysis. The guidance weights for the lowest MAE of AG C_v are $(w_1, w_2) = (3, 1)$ with an MAE of 0.654 cal/(mol·K), which is slightly worse than the result (0.638 cal/(mol·K)) using the guidance weights $(w_1, w_2) = (2.75, 1.14)$.

Ablation study on the dependence of molecule stability on guidance weights for MG can be found in Table 17.

Results for guidance on four weights versus guidance on two weights for AG can be found in Table 18

G. Additional Details for the Methods

QM9 Properties and Data Details.

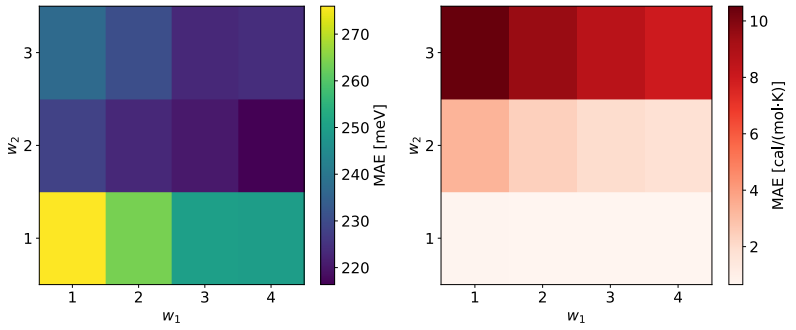

Figure 9: Effects of guidance weights on the property alignment for CFG ϵ_{HOMO}(Left) and AG C_v(Right).

Table 17: Molecule Stability for MG versus guidance weights. This molecule stability and bond distance std is averaged across models conditioned on six properties.

w	Molecule Stability [%]	Bond Entropy	MAE, $\Delta\epsilon$[meV]
1	95.1 ± 2.1	0.536 ± 0.044	429
2	95.0 ± 2.1	0.535 ± 0.044	427
3	95.1 ± 2.3	0.533 ± 0.044	429
4	95.1 ± 2.1	0.536 ± 0.048	429

Table 18: Comparison of property MAEs with four guidance weights against two guidance weights for AG.

AG	α	$\Delta\epsilon$	ϵ_{HOMO}	ϵ_{LUMO}	μ	C_v
Two weights	1.43	344	242	274	0.631	0.638
Four weights	1.46	341	247	265	0.620	0.640

- α (Polarizability): Tendency of a molecule to acquire an eletric dipole moment when subjected to an external electric field.
- $\Delta\epsilon$: The energy gap between HOMO and LUMO.
- ϵ_{HOMO}: Highest occupied molecule orbital energy.
- ϵ_{LUMO}: Lowest unoccupied molecule orbital energy.
- μ: Dipole moment, which measures the separation of positive and negative charges within a molecule.
- C_v: Heat capacity at room temperature 298.15 K.

QM9 is a 134k small molecule dataset that only contains of up to 9 heavy atoms (C, N, O, F). The atom sizes range from 3 to 29 with an average of 18 atoms, including explicit hydrogen. All molecules are optimized by density functional theory (DFT) calculations and thus in their stable states. By design, all molecules in QM9 are charge neural and have a close shell valence electron configuration. But one should note there are molecules that carry explicit atom formal charges and a molecule graph carrying this information might be hence helpful—like we did in this work—to generate molecules with valid charge–valency configuration.

rQM9 SDF data The original QM9 SDF data can be retrieved from DeepChem and has bond and charge issues. We noticed this in our previous work and have corrected all invalid bond orders and charges. The procedure to correct the data is provided in Appendix of (Zeng et al., 2025b). The data is now available at HuggingFace ColabFit (Zeng et al., 2025a).

Training details and hyperparamters. All guidance models on the QM9 dataset 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 (Kingma & Ba, 2017) 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 input to predict the target property. The parameters of GVP regressors are optimized by minimizing an mean squared error loss function. Separate models were trained for each of the six tested properties. The GVP training uses a different 50k QM9 data compared to the 50k used for the PropMolFlow model training.

Definition of metrics in the radar plot. Below we elaborate the definition of each metric in the radar plot (Figure 4). If a higher value is preferred (*e.g.*, Bond Diversity, Structure Validity), the metric is transformed via the Eq. 22 .

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{22}$$

If a lower value (*e.g.*, Sampling Efficiency by time, Property Alignment by MAEs) is preferred, the following linear transformation is used.

$$x' = \frac{x_{\max} - x}{x_{\max} - x_{\min}} \tag{23}$$

Sampling efficiency follows the same definition and the min and max scaling factors are 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 2.