# A Proofs of Propositions

In this section, we prove the properties listed in § 4.2.

The PDF of the Bingham distribution is rewritten as

$$p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda}) = \frac{1}{Z(\boldsymbol{\Lambda})} \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{q}\right).$$

**Proposition 1.** *The PDF of the Bingham distribution maintains the antipodal symmetry,* i.e.*,* $p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda}) = p_B(-\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda})$.

*Proof.*

$$\begin{aligned} p_B(-\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda}) &= \frac{1}{Z(\boldsymbol{\Lambda})} \exp\left((-\boldsymbol{q}^\top) \boldsymbol{M}^\top \boldsymbol{\Lambda} \boldsymbol{M} (-\boldsymbol{q})\right) \\ &= \frac{1}{Z(\boldsymbol{\Lambda})} \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{q}\right) \\ &= p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda}) \end{aligned}$$

□

**Proposition 2.** *When* $\boldsymbol{\Lambda} = \mathbf{0}$*, the PDF is reduced to a uniform distribution,* i.e.*,* $p_B(\boldsymbol{q}; \boldsymbol{M}, \mathbf{0}) \equiv \frac{1}{2\pi^2}$.

*Proof.* When $\boldsymbol{\Lambda} = \mathbf{0}$, the exponent becomes zero for all $\boldsymbol{q} \in \mathcal{S}^3$:

$$p_B(\boldsymbol{q}; \boldsymbol{M}, \mathbf{0}) = \frac{1}{Z(\mathbf{0})} \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top \mathbf{0} \boldsymbol{M} \boldsymbol{q}\right) = \frac{1}{Z(\mathbf{0})}.$$

That is, $p_B(\boldsymbol{q}; \boldsymbol{M}, \mathbf{0})$ is constant over $\mathcal{S}^3$, indicating it is a uniform distribution. Since the surface area of $\mathcal{S}^3$ is $2\pi^2$, the normalized uniform density is $p_B(\boldsymbol{q}; \boldsymbol{M}, \mathbf{0}) \equiv \frac{1}{2\pi^2}$. □

**Proposition 3.** *Due to the normalization constraint on the hypersphere, any bias applied on the eigenvalues would not affect the distribution,* i.e.*,* $p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda} + k\boldsymbol{I}) = p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda}), \forall k \in \mathbb{R}$.

*Proof.*

$$\begin{aligned} p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda} + k\boldsymbol{I}) &= \frac{1}{Z(\boldsymbol{\Lambda} + k\boldsymbol{I})} \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top (\boldsymbol{\Lambda} + k\boldsymbol{I}) \boldsymbol{M} \boldsymbol{q}\right) \\ &\propto \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top (\boldsymbol{\Lambda} + k\boldsymbol{I}) \boldsymbol{M} \boldsymbol{q}\right) \\ &= \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{q} + k \boldsymbol{q}^\top \boldsymbol{M}^\top \boldsymbol{M} \boldsymbol{q}\right) \\ &= e^k \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{q}\right) \\ &\propto \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{q}\right). \end{aligned}$$

Given the normalization constant $Z(\boldsymbol{\Lambda})$, we have $p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda}+k\boldsymbol{I}) = \frac{1}{Z(\boldsymbol{\Lambda})} \exp\left(\boldsymbol{q}^\top \boldsymbol{M}^\top \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{q}\right) = p_B(\boldsymbol{q}; \boldsymbol{M}, \boldsymbol{\Lambda})$. □

# B Implementation Details for MOF-BFN

## B.1 Determining Local Geometries for Building Blocks

To determine the orientations of building blocks, we should first determine a reference frame for each block. A common-used solution is Principle Component Analysis (PCA) [4]. Given a building block $\mathcal{C}_j = (\boldsymbol{A}_j, \boldsymbol{X}_j)$, where $\boldsymbol{A}_j = [\boldsymbol{a}_r]_{r=1}^{N_j} \in \mathbb{R}^{h \times N_j}, \boldsymbol{X}_j = [\boldsymbol{x}_r]_{r=1}^{N_j} \in \mathbb{R}^{3 \times N_j}$ denote the atom types and coordinates within the block. The consistent local structure $\dot{\boldsymbol{X}}_j$ is defined as

$$\dot{\boldsymbol{X}}_j = \tilde{\boldsymbol{R}} \bar{\boldsymbol{X}}_j, \tag{28}$$

where $\bar{\boldsymbol{X}}_j = [\bar{\boldsymbol{x}}_r]_{r=1}^{N_j} = [\boldsymbol{x}_r - \frac{1}{N_j}\sum_{r=1}^{N_j}\boldsymbol{x}_r]_{r=1}^{N_j}$ is the centered coordinates, and $\tilde{\boldsymbol{R}} = \boldsymbol{c} \odot \boldsymbol{R}$. $\boldsymbol{R} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$ is the eigenvector matrix of $\bar{\boldsymbol{X}}_j\bar{\boldsymbol{X}}_j^\top$, and the signs $\boldsymbol{c} = [c_1, c_2, c_3]$ is determined by

$$c_i = \begin{cases} +1, \boldsymbol{e}_i^\top \boldsymbol{v}_j \geq 0, \\ -1, \boldsymbol{e}_i^\top \boldsymbol{v}_j < 0, \end{cases} \tag{29}$$

with a vector $\boldsymbol{v}_j$ determining the direction of the building block. Following [6], the vector is defined as $\boldsymbol{v}_j = \arg\min_{\bar{\boldsymbol{x}}_r, \|\bar{\boldsymbol{x}}_r\|\neq 0} \|\bar{\boldsymbol{x}}_r\|$.

### B.2 Rejection Sampling for Bingham Distribution

To sample from the Bingham distribution, we adopt a rejection sampling strategy using the Angular Central Gaussian (ACG) distribution as the proposal [5]. Given a multivariate normal distribution $\boldsymbol{u} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, the normalized direction $\boldsymbol{q} = \boldsymbol{q}/\|\boldsymbol{q}\|$ follows an ACG distribution, whose density is $p_{\text{ACG}}(\boldsymbol{q}) \propto (\boldsymbol{q}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{q})^{-d/2}$, where $d$ is the dimension of the hypersphere. To perform rejection sampling, we choose $\boldsymbol{\Sigma}^{-1} = \boldsymbol{I} - 2\boldsymbol{\Lambda}$ such that the Bingham distribution is upper bounded by a scaled ACG distribution: $p_B(\boldsymbol{q}) \leq A' \cdot p_{\text{ACG}}(\boldsymbol{q})$, with $A'$ related to the normalization term of each distribution. This bound leads to the condition

$$\begin{aligned} \exp(\boldsymbol{q}^\top\boldsymbol{\Lambda}\boldsymbol{q}) &\leq A(\boldsymbol{q}^\top(\boldsymbol{I} - 2\boldsymbol{\Lambda})\boldsymbol{q})^{-d/2} \\ A &\geq \frac{\exp(\boldsymbol{q}^\top\boldsymbol{\Lambda}\boldsymbol{q})}{(\boldsymbol{q}^\top(\boldsymbol{I} - 2\boldsymbol{\Lambda})\boldsymbol{q})^{-d/2}} \\ &= \exp(\boldsymbol{q}^\top\boldsymbol{\Lambda}\boldsymbol{q})(1 - 2\boldsymbol{q}^\top\boldsymbol{\Lambda}\boldsymbol{q})^{d/2} \end{aligned}$$

We define the function $f(t) = \exp(t)(1-2t)^{d/2}$ and maximize it over $t \in \mathbb{R}$ to obtain the tightest possible rejection bound. The maximum occurs at $t = \frac{1-d}{2}$, resulting in an optimal rejection constant $A_{\min} = \exp(\frac{1-d}{2})d^{d/2}$. Once a sample $\boldsymbol{q}$ is accepted, we apply a linear transformation $\boldsymbol{q}' = \boldsymbol{M}\boldsymbol{q}$ if the Bingham distribution has eigendecomposition $\boldsymbol{M}\boldsymbol{\Lambda}\boldsymbol{M}^\top$. This yields a sample from the desired Bingham distribution.

### B.3 Accuracy Scheduling for Bingham BFN

As the eigenvalue matrix $\boldsymbol{\Lambda}$ is diagonal, we denote the normalization term $Z(\boldsymbol{\Lambda})$ as $Z(\boldsymbol{\lambda})$ in this subsection. To determine a suitable value of $\alpha_i$, we consider the entropy of the Bingham distribution, which takes the form

$$H_i = H(\boldsymbol{\lambda}_i) = \log Z(\boldsymbol{\lambda}_i) - \boldsymbol{\lambda}_i^\top \nabla \log Z(\boldsymbol{\lambda}_i),$$

Assuming that $\boldsymbol{\lambda}_i \approx [0, -\beta_i, -\beta_i, -\beta_i]$, we can approximate the entropy as a function of a single parameter $\beta_i$. This assumption is an approximation that enforces isotropy around the principal axis, but it is effective in practice for constructing a simple and stable entropy scheduler. Let $\beta_0 = 0$ and $\beta_T$ sufficiently large, we linearly interpolate the entropy as $H_i = (1-\frac{i}{T})H_0 + \frac{i}{T}H_T$, and numerically solve for each intermediate value $\beta_i$. Consider the sender distribution $\boldsymbol{y}_i \sim p_W([1,0,0,0], \alpha_i)$, its second moment is given by

$$\mathbb{E}[\boldsymbol{y}_i\boldsymbol{y}_i^\top] = \text{diag}\Big(\frac{\nabla Z(0, -\alpha_i, -\alpha_i, -\alpha_i)}{Z(0, -\alpha_i, -\alpha_i, -\alpha_i)} + \big(1 - \sum \frac{\nabla Z(0, -\alpha_i, -\alpha_i, -\alpha_i)}{Z(0, -\alpha_i, -\alpha_i, -\alpha_i)}\big)[1,0,0,0]\Big).$$

From this, the expected change in $\beta$ is

$$\begin{aligned} \mathbb{E}[\beta_i - \beta_{i-1}] &= \alpha_i(\mathbb{E}[\boldsymbol{y}_i\boldsymbol{y}_i^\top]_0 - \mathbb{E}[\boldsymbol{y}_i\boldsymbol{y}_i^\top]_1) \\ &= \alpha_i\big(1 - 4\frac{\nabla Z(0, -\alpha_i, -\alpha_i, -\alpha_i)_1}{Z(0, -\alpha_i, -\alpha_i, -\alpha_i)}\big), \end{aligned}$$

where the subscripts indicate the value at the corresponding indices. And we can numerically solve for $\alpha_i$ given $\beta_i$. Such scheduler gradually sharpens the Bingham distribution, increasing its concentration around the target direction, ensuring that the model starts with a high-entropy prior and becomes progressively confident as step increasing.

# C Extension to De Novo Generation

## C.1 Implementation Details

The utilized MOF datasets contains millions of building block conformations, making a simple one-hot encoding for building blocks sparse and computationally inefficient. To obtain compact and continuous representations, a contrastive learning framework is employed in MOFDiff [3]. Each building block is encoded into a latent vector using a SE(3)-equivariant message passing neural network, specifically GemNet-OC.

Let $\mathcal{G}$ denote the set of all building blocks obtained from training MOFs. For each building block $\mathcal{C} \in \mathcal{G}$, the encoder network $f_\theta$ maps its local structure to a continuous embedding $\mathbf{z}_\mathcal{C} = f_\theta(\mathcal{C}) \in \mathbb{R}^d$, with latent dimension $d = 32$ in MOFDiff.

To ensure that the learned embeddings reflect chemical similarity, contrastive learning is performed using positive and negative pairs of building blocks. A positive pair $(\mathcal{C}, \mathcal{C}^+)$ consists of two blocks sharing the same ECFP4 fingerprint, while a negative pair $(\mathcal{C}, \mathcal{C}^-)$ indicates different block identities.

The contrastive learning objective is based on the following loss:

$$\mathcal{L}_{\text{contrast}} = -\sum_{i \in \mathcal{S}} \log \frac{\sum_{j \in \mathcal{S}+} \exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}{\sum_{j \in \mathcal{S}} \exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}, \tag{30}$$

where $\mathcal{S}, \mathcal{S}+$ denote the batch and the positive subset of the batch, $\tau > 0$ is a temperature hyperparameter, and $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity:

$$\text{sim}(\mathbf{z}_1, \mathbf{z}_2) = \frac{\mathbf{z}_1^\top \mathbf{z}_2}{\|\mathbf{z}_1\| \, \|\mathbf{z}_2\|}, \tag{31}$$

While the continuous encoding space enables efficient retrieval via KD-Trees, we empirically observe that the distribution of embeddings deviates significantly from a normal distribution, which poses challenges for training a BFN. To address this, we normalize the representations as

$$\bar{\boldsymbol{z}}_i = \frac{\boldsymbol{z}_i - \text{mean}_{j \in \mathcal{G}}(\boldsymbol{z}_j)}{\text{std}_{j \in \mathcal{G}}(\boldsymbol{z}_j)}.$$

Similar to the BFN for lattice parameters, given the block embeddings $\bar{\boldsymbol{Z}} = [\bar{\boldsymbol{z}}_i]_{i=1}^K$ of a structure, the input distribution is given by $\mathcal{N}(\bar{\boldsymbol{Z}}; \boldsymbol{\mu}_i^\mathcal{B}, (\rho_i^\mathcal{B})^{-1}\boldsymbol{I})$ parameterized by $\boldsymbol{\theta}_i^\mathcal{B} = \{\boldsymbol{\mu}_i^\mathcal{B}, \rho_i^\mathcal{B}\}$. and the prior distribution is chosen as $\boldsymbol{\theta}_0^\mathcal{B} = \{\mathbf{0}, 1\}$. After acquiring a sample from the sender distribution $\boldsymbol{y}_i^\mathcal{B} \sim \mathcal{N}(\bar{\boldsymbol{Z}}, (\alpha_i^\mathcal{B})^{-1}\boldsymbol{I})$ at step $i$, the Bayesian update function is

$$\{\boldsymbol{\mu}_i^\mathcal{B}, \rho_i^\mathcal{B}\} = \Big\{\frac{\rho_{i-1}^\mathcal{B}\boldsymbol{\mu}_{i-1}^\mathcal{B} + \alpha_i^\mathcal{B}\boldsymbol{y}_i^\mathcal{B}}{\rho_{i-1}^\mathcal{B} + \alpha_i^\mathcal{B}}, \rho_{i-1}^\mathcal{B} + \alpha_i^\mathcal{B}\Big\}. \tag{32}$$

The corresponding Bayesian flow distribution is accumulated as

$$p_F^\mathcal{B}(\boldsymbol{\mu}_i^\mathcal{B} | \bar{\boldsymbol{Z}}, i) = \mathcal{N}\big((1 - \sigma_T^{2i/T})\bar{\boldsymbol{Z}}, \sigma_T^{2i/T}(1 - \sigma_T^{2i/T})\boldsymbol{I}\big). \tag{33}$$

The training objective on the latent space is

$$\mathcal{L}_\mathcal{B} = \mathbb{E}_{i \sim U(1,T), \boldsymbol{\mu}_{i-1}^\mathcal{B} \sim p_F^\mathcal{B}(\boldsymbol{\mu}_{i-1}^\mathcal{B} | \bar{\boldsymbol{Z}}, i-1)} \Big[\frac{\alpha_i^\mathcal{B} T}{2} \|\bar{\boldsymbol{Z}} - \phi_\mathcal{B}(\boldsymbol{\theta}_{i-1}^\mathcal{M}, i)\|_2^2\Big]. \tag{34}$$

And the entire training objective is extended as

$$\mathcal{L}_{\text{DNG}} = \lambda_{\boldsymbol{\xi}}\mathcal{L}_{\boldsymbol{\xi}} + \lambda_{\boldsymbol{F}}\mathcal{L}_{\boldsymbol{F}} + \lambda_{\boldsymbol{q}}\mathcal{L}_{\boldsymbol{q}} + \lambda_\mathcal{B}\mathcal{L}_\mathcal{B}.$$

## C.2 Additional Results

Similar to MOFDiff, we further relax the generated structures via the UFF force field [7]. We refine both the lattice parameters and the all-atom positions by `LAMMPS` [8] and `LAMMPS Interface` [2]. The numbers of valid structures before and after relaxation are reported in Table 4, further demonstrating the generation quality of MOF-BFN.

Table 4: **Generation validity.** Number of structures that passed (↑) or failed (↓) each criterion among 1,000 generated candidates.

| | Before Relaxation | | After Relaxation | |
|---|---|---|---|---|
| Validity Criteria | MOF-BFN | MOFDiff | MOF-BFN | MOFDiff |
| *Connection Point Matching* | | | | |
| matched ↑ | **923** | 723 | **923** | 723 |
| *UFF Relaxation* | | | | |
| relaxed ↑ | - | - | **849** | 662 |
| *MOFChecker* | | | | |
| has_carbon ↑ | **923** | 723 | **923** | 723 |
| has_hydrogen ↑ | **900** | 715 | **827** | 654 |
| has_atomic_overlaps ↓ | **106** | 187 | **35** | 136 |
| has_overcoordinated_c ↓ | **178** | 264 | **8** | 17 |
| has_overcoordinated_n ↓ | 45 | **34** | **0** | **0** |
| has_overcoordinated_h ↓ | **166** | 247 | **20** | 21 |
| has_undercoordinated_c ↓ | **168** | 179 | **144** | 194 |
| has_undercoordinated_n ↓ | 142 | **91** | 141 | **133** |
| has_undercoordinated_rare_earth ↓ | **0** | **0** | **0** | **0** |
| has_metal ↑ | **923** | 723 | **923** | 723 |
| has_lone_molecule ↓ | **172** | 316 | **36** | 60 |
| has_high_charges ↓ | **64** | 104 | **5** | 13 |
| has_suspicious_terminal_oxo ↓ | **0** | 1 | **0** | 2 |
| has_undercoordinated_alkali_alkaline ↓ | 25 | **1** | **0** | **0** |
| has_geometrically_exposed_metal ↓ | 281 | **253** | **8** | 24 |
| *Total* | | | | |
| total_valid ↑ | **323** | 107 | **545** | 317 |

# D Experimental Details

Hyperparameters for the structure prediction (§ 5.1) and de novo generation (§ 5.3) are provided in Table 5. Baseline results in Table 1 and 2 are from MOFFlow [6], and the samples for calculating MOFDiff results in Table 3 are directly yielded from the official pre-trained checkpoint [2]. To ensure a fair comparison, we share the same split with MOFDiff, where 95% structures are used for training, and the remaining 5% are for validation. The structure prediction and de novo generation models are trained on 8 GPUs with 80 GB memories, and the training procedures take 136 and 152 GPU hours, respectively.

# E Limitations

While MOF-BFN presents a promising approach to hierarchical MOF structure prediction, several limitations remain that we leave to future work. First, our current framework treats each building block as a rigid body with a fixed local geometry. This rigid-body assumption simplifies the generative process but neglects the intrinsic conformational flexibility of many organic linkers and secondary building units. For instance, MOFDiff [3] reports that over 2 million building block instances in the dataset correspond to only 242k unique molecular graphs, indicating that significant conformational diversity exists within each block type. Although our method supports the extension of building block vocabularies through a continuous embedding space, it does not yet account for conformation generation within each block. Integrating internal flexibility modeling into the current framework could further enhance the model. Second, our current work focuses on unconditional generation and structure prediction tasks, without explicitly incorporating guidance signals for specific downstream properties. In practice, many MOF design scenarios require property-oriented generation, such as optimizing for gas adsorption capacity or catalytic activity. However, techniques for guiding models toward desired property targets remain underexplored in the field of Bayesian Flow Networks.

[2] https://github.com/microsoft/MOFDiff

Table 5: Hyperparameter settings for experiments.

| *Building Block Encoder* | | | | |
|---|---|---|---|---|
| num_layers | node_dim | edge_dim | hidden_dim | max_radius |
| 4 | 64 | 64 | 64 | 5 |
| *Coarse-Grained Structure Predictor* | | | | |
| num_layers | hidden_dim | time_dim | num_freqs | |
| 6 | 512 | 128 | 64 | |
| *BFN* | | | | |
| $\beta_T^{\boldsymbol{\xi}}$ | $\beta_T^{\boldsymbol{F}}$ | $\beta_T^{\boldsymbol{q}}$ | $\beta_T^{\mathcal{B}}$ | $T$ |
| 1000 | 1000 | 200 | 1000 | 50 |
| $\lambda_{\boldsymbol{\xi}}$ | $\lambda_{\boldsymbol{F}}$ | $\lambda_{\boldsymbol{q}}$ | $\lambda_{\mathcal{B}}$ | |
| 1.0 | 1.0 | 0.2 | 10.0 | |
| *Structure Prediction Training* | | | | |
| lr | min_lr | plateau_factor | plateau_patience | Adam_betas |
| $5\times10^{-4}$ | $1\times10^{-4}$ | 0.6 | 30 | [0.9, 0.98] |
| epochs | batch_size | gradient_clip_val | weight_decay | |
| 1000 | 512 | 0.5 | 0.01 | |
| *De Novo Generation Training* | | | | |
| lr | min_lr | plateau_factor | plateau_patience | Adam_betas |
| $5\times10^{-4}$ | $1\times10^{-4}$ | 0.6 | 30 | [0.9, 0.98] |
| epochs | batch_size | gradient_clip_val | weight_decay | |
| 3000 | 512 | 0.5 | 0.01 | |

Developing conditional generation mechanisms within the BFN framework is an important direction for future research to enable targeted material discovery.

# F Broader Impacts

This work contributes to the development of MOF structure prediction and design. It may benefit applications in gas storage, separation, and catalysis by enabling more efficient exploration of the chemical design space. By improving structure prediction accuracy and generation validity, it can potentially accelerate material discovery in a data-driven way. However, our model is trained and evaluated primarily on the BW-DB dataset [1], which may contain inherent biases in block types, structural motifs, or chemical compositions. As a result, the generalization ability of the model to underrepresented MOF types or application-specific domains could be limited. Care should be taken when applying the model beyond the scope of the training data.

# G Code Availablility

Our codes are available at `https://anonymous.4open.science/r/MOF-BFN`.

# References


[1] Peter G Boyd, Yongjin Lee, and Berend Smit. Computational development of the nanoporous materials genome. *Nature Reviews Materials*, 2(8):1–15, 2017.

[2] Peter G Boyd, Seyed Mohamad Moosavi, Matthew Witman, and Berend Smit. Force-field prediction of materials properties in metal-organic frameworks. *The journal of physical chemistry letters*, 8(2):357–363, 2017.

[3] Xiang Fu, Tian Xie, Andrew Scott Rosen, Tommi S Jaakkola, and Jake Allen Smith. Mofdiff: Coarse-grained diffusion for metal-organic framework design. In *The Twelfth International Conference on Learning Representations*.

[4] Nicholas Gao and Stephan Günnemann. Ab-initio potential energy surfaces by pairing gnns with neural wave functions. In *International Conference on Learning Representations*.

[5] John T Kent, Asaad M Ganeiber, and Kanti V Mardia. A new method to simulate the bingham and related distributions in directional data analysis with applications. *arXiv preprint arXiv:1310.8110*, 2013.

[6] Nayoung Kim, Seongsu Kim, Minsu Kim, Jinkyoo Park, and Sungsoo Ahn. Mofflow: Flow matching for structure prediction of metal-organic frameworks. *arXiv preprint arXiv:2410.17270*, 2024.

[7] Anthony K Rappé, Carla J Casewit, KS Colwell, William A Goddard III, and W Mason Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992.

[8] Aidan P Thompson, H Metin Aktulga, Richard Berger, Dan S Bolintineanu, W Michael Brown, Paul S Crozier, Pieter J In't Veld, Axel Kohlmeyer, Stan G Moore, Trung Dac Nguyen, et al. Lammps-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer physics communications*, 271:108171, 2022.