# CoFM: Molecular Conformation Generation via Flow Matching in SE(3)-Invariant Latent Space

**Guikun Xu** [*1] **Yankai Yu** [*1] **Yongquan Jiang** [1] **Yan Yang** [1] **Yatao Bian** [2]

## Abstract

Current leading methods for molecular conformation generation often rely on computationally intensive diffusion models in 3D space, which struggle with accurately modeling conformational manifolds and rigorously maintaining SE(3) equivariance. These limitations hinder both performance and efficiency, and can complicate integration with standard tools like RDKit. To overcome these challenges, we introduce **CoFM**, a novel generative framework that pioneers the concept of an autoencoder-induced, fully SE(3)-invariant latent space. This approach decouples SE(3) equivariance constraints from the generation process, enabling seamless integration of RDKit's physicochemical priors. Furthermore, **CoFM** is the first to integrate latent flow matching within this invariant geometric subspace, significantly enhancing generation efficacy with fewer iterative steps. Experimental validation demonstrates that our method generates high-quality results with fewer iterations, achieving significant improvements in key *Precision* metrics and ensuring greater energy authenticity.

## 1. Introduction

*Molecular Conformation Generation*, which involves deriving low-energy stable conformations from molecular graphs, is a key challenge in drug discovery and bioinformatics. Traditional methods, such as those based on handcrafted force fields (Rappé et al., 1992; Halgren, 1996) or density functional theory (Parr et al., 1979), often suffer from limited precision or high computational cost. Recently, deep generative models have emerged as the dominant approach, driven by advancements in the deep learning community.

These methods can be generally divided into two categories. Early studies (Simm & Hernández-Lobato, 2019; Xu et al., 2021b;a) focus on modeling interatomic distances by generating atomic distance matrices with VAE or Flow, followed by distance geometry optimization (Liberti et al., 2014) to derive conformations. ConfGF (Shi et al., 2021) attempts to build a Noise Conditional Score Network (NCSN) (Song & Ermon, 2019) on conformations, but it uses the chain rule to shift the estimation of the conformation's score to the atomic distance matrix.

Although some progress has been made, the above methods are primarily constrained by the inherent limitations of modeling atomic distance matrices, as recovering conformations from these matrices is difficult and inefficient, often failing to restore the 3D structure. In light of these challenges, and considering the robust capabilities of diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) along with their successful applications in other domains (Dhariwal & Nichol, 2021; Rombach et al., 2022), several works have emerged that directly construct diffusion models in 3D conformation space. GeoDiff (Xu et al., 2022) successfully integrates a discrete diffusion model (Ho et al., 2020) into molecular conformation generation in coordinate space, yielding significant results. Following this, SDEGen (Zhang et al., 2023) extends the methodology to diffusion models based on SDE (Song et al., 2020), while EC-Conf (Fan et al., 2023) implements a consistency model (Song et al., 2023) rooted in ODE to accelerate generations. Additionally, TorDiff (Jing et al., 2022) builds diffusion on the torsional angle hypersurface of coarse RDKit-based conformations. MCF (Wang et al., 2023) trains a diffusion model that maps elements from the molecular graph to points in 3D space, whereas ETFlow (Hassan et al., 2024) trains flow matching to align harmonic prior with the target conformation space.

While methods for constructing diffusion models in the 3D conformational space have achieved significant performance gains, they have established themselves as mainstream and robust approaches while still facing inherent modeling challenges that create performance bottlenecks. Firstly, the conformational space of molecules is highly nonlinear and

[*]Equal contribution [1]School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China [2]National University of Singapore, Singapore. Correspondence to: Yongquan Jiang <yqjiang@swjtu.edu.cn>, Yatao Bian <bianyt@comp.nus.edu.sg>.

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

often forms complex manifolds, which makes accurately capturing and modeling the distribution a significant challenge. Secondly, molecular conformations exhibit SE(3) equivariance in 3D space, requiring that the corresponding likelihood (conditioned on the molecular graph) remains SE(3)-invariant – remaining unaffected by rotational and translational transformations. This fundamental requirement underscores the importance of establishing both an SE(3)-invariant prior distribution for sampling procedures and an SE(3)-equivariant Markov kernel for diffusion model construction (Xu et al., 2022; Zhang et al., 2023; Fan et al., 2023). Otherwise, directly sampling from a 3D Gaussian prior poses challenges for most methods in seamlessly integrating tools like RDKit (Landrum, 2006), thereby limiting the accuracy of the generated conformations. Lastly, the multi-step denoising process of diffusion models significantly reduces their sampling efficiency in practical applications. Consequently, these stringent conditions significantly amplify problem complexity, presenting formidable obstacles in designing diffusion models that simultaneously achieve high performance and effectiveness in conformation generation.

To address these significant challenges, we present **CoFM** (**Co**nformational Latent **F**low **M**atching, Figure 1), an innovative solution inspired by recent advances in latent diffusion paradigms (Rombach et al., 2022; Esser et al., 2024) and flow matching techniques (Lipman et al., 2022; Liu, 2022). Our primary technical contribution involves developing an advanced autoencoder framework that directly ***induces a fully SE(3)-invariant latent space***. This framework facilitates seamless integration with RDKit's physicochemical priors while rigorously decoupling SE(3) equivariance from subsequent diffusion processes, thereby enabling the diffusion model to more effectively and precisely learn the distribution of geometric features. Furthermore, we pioneer the implementation of flow matching within this novel SE(3)-invariant latent space, achieving substantial improvements in both generation efficiency and conformational reliability for molecular systems. **The highlights of this paper are:**

- Development of an advanced autoencoder framework that establishes a fully SE(3)-invariant latent space. This framework effectively incorporates RDKit's robust physicochemical priors while systematically decoupling SE(3) equivariance from the diffusion process. By entirely delegating equivariance constraints to the autoencoder, the diffusion model can focus exclusively on learning latent geometric feature distributions.
- First application of flow matching to molecular conformation generation in non-coordinate space. This innovation substantially improves both the efficiency of conformation sampling and the physicochemical reliability of generated structures.
- Introduction of **CoFM**, a novel algorithm for molecular conformation generation. Comprehensive experiments validate the method's superiority in critical precision metrics, demonstrating significant enhancements in generation speed and the energetic attributes of produced conformational ensembles.

## 2. Related Work

Recently, generative models have seen growing adoption for generating multiple molecular conformations. CGCF (Xu et al., 2021a) uses a flow model to capture the distribution of interatomic distances $\mathcal{D}$ given a molecular graph $\mathcal{G}$. Conformations are generated from $p(\mathcal{C}|\mathcal{D}, \mathcal{G})$ and refined with a Markov Chain Monte Carlo (MCMC) procedure guided by an Energy-based Tilting Model. ConfVAE (Xu et al., 2021b) employs a bilevel programming framework, which divides the task into distance prediction and distance geometry optimization. A conditional variational autoencoder (VAE) predicts interatomic distances conditioned on molecular graphs, while 3D conformations are reconstructed from these distances by distance geometry optimization (Liberti et al., 2014). ConfGF (Shi et al., 2021) proposes first estimating the gradient field of interatomic distances and then deriving the gradient field of the log density (i.e., scores) of atomic coordinates via the chain rule. The conformations are subsequently sampled using an annealed Langevin dynamics algorithm based on the estimated scores. However, these methods rely on modeling interatomic distances as an intermediate variable to generate conformational coordinates, a factor identified as contributing to their suboptimal performance. To overcome the limitations of previous approaches, recent work has shifted toward direct modeling in 3D coordinate space. GeoMol (Ganea et al., 2021) focuses on key geometric features of molecules, such as torsion angles, bond lengths, and bond angles, and generates these elements during inference to reconstruct complete 3D conformations. DMCG (Zhu et al., 2022) is the first attempt to construct a carefully designed VAE framework directly in a 3D space, achieving promising results. Moreover, numerous studies have begun to take advantage of the increasingly popular diffusion models (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020; 2023). GeoDiff (Xu et al., 2022) implements the diffusion processes directly on the atomic coordinates, recovering the desired conformation from positions sampled from the noise. TorDiff (Jing et al., 2022) is the first to leverage RDKit's prior knowledge, restricting diffusion to the torsional hypersurface of RDKit-generated molecular conformations. Recent research (Zhang et al., 2023; Fan et al., 2023) has focused on reducing inference steps by introducing newly proposed diffusion model acceleration algorithms (Song et al., 2020; 2023; Albergo et al.,

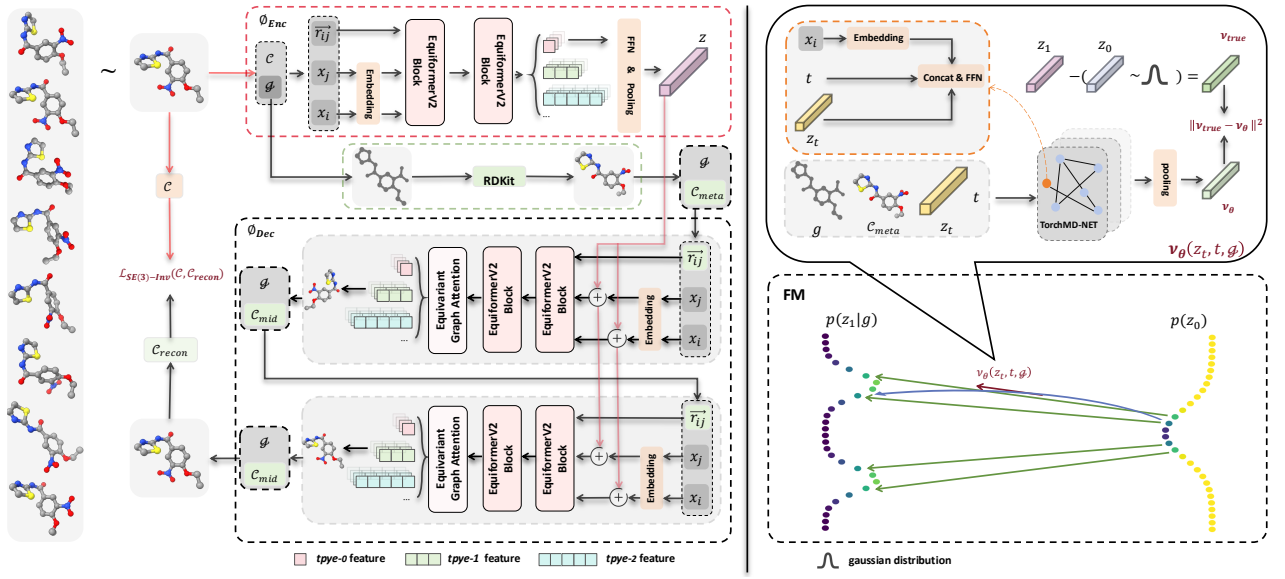



*Figure 1.* **Overview of the CoFM Framework. Left:** The autoencoder constructs a fully *SE(3)-invariant latent space* while seamlessly integrating RDKit's prior knowledge. **Right:** The *flow matching* algorithm operates within the latent space to generate new latent representations. The **upper-right** section illustrates the vector field predictor $\boldsymbol{v}_\theta$, while the **lower-right** section depicts the *latent flow matching Algorithm*, which maps the prior noise distribution $p(\mathbf{z}_0) \sim \mathcal{N}(0, I)$ to the target data distribution $p(\mathbf{z}|\mathcal{G})$.

2023). ETFlow (Hassan et al., 2024) defines the source distribution as the Harmonic Prior and the target distribution as the true conformation space, then uses flow matching (Albergo & Vanden-Eijnden, 2022) to connect them. These methods result in significant improvements in both precision and speed, while still operating within coordinate space.

## 3. Background

### 3.1. Notations and Problem Definition

**Main Notations.** A 3D molecule with $n$ atoms (nodes) and $m$ bonds (edges) is represented as $\mathbf{G} = \{\mathbf{x}, \mathbf{E}, \mathcal{C}\}$. Here, $\mathbf{x} \in \mathbb{N}^n$ denotes the atom types, where $x_i$ specifies the type of the $i$-th node. The connectivity of the $m$ edges is given by $\mathbf{E} \in \mathbb{N}^{m\times 2}$, where $\mathbf{E}_k = [i, j]$ for $1 \leq k \leq m$ indicates an edge between nodes $i$ and $j$. The 3D coordinates of these nodes are denoted as $\mathcal{C} \in \mathbb{R}^{n\times 3}$, with $\mathcal{C}_i$ representing the coordinates of the $i$-th node. The relative vector between nodes $i$ and $j$ is $\vec{r}_{ij} = \mathcal{C}_j - \mathcal{C}_i$, and the squared Euclidean distance is $\mathcal{D}_{ij} = ||\mathcal{C}_j - \mathcal{C}_i||$, where $\mathcal{D} \in \mathbb{R}^{n\times n}$ is the pairwise distance matrix. The molecular topology, excluding conformational information, is represented as $\mathcal{G} = \{\mathbf{x}, \mathbf{E}\}$. Additionally, $\mathbf{D}$ represents the dataset, while $\mathbf{S}_r$ and $\mathbf{S}_g$ denote the reference and generated sets of conformations, respectively. Other relevant notations are introduced in their respective sections.

**Problem Definition.** The task of ***molecular conformation generation*** aims to produce an ensemble of multiple potential low-energy stable conformations, $\{\mathcal{C}^1, \mathcal{C}^2, \ldots\}$, for a given molecular graph $\mathcal{G}$. Thus, our objective is to learn a conditional distribution $p(\mathcal{C}|\mathcal{G})$.

### 3.2. SE(3) Equivariance/Invariance in Atomic Systems

Formally, given two vector spaces $X$ and $Y$, let $D_X(g)$ and $D_Y(g)$ denote the transformation matrices parameterized by $g \in SE(3)$ in $X$ and $Y$, respectively. For any input $x \in X$, output $y \in Y$, a mapping $f : X \to Y$ is considered SE(3)-equivariant if:

$$f(D_X(g)x) = D_Y(g)f(x). \tag{1}$$

Similarly, a mapping $f : X \to Y$ is considered SE(3)-invariant if:

$$f(D_X(g)x) = f(x). \tag{2}$$

For atomic systems represented in 3D Cartesian space (e.g., conformational molecules), the concepts of equivariance and invariance under SE(3), the group of rigid transformations comprising translations and rotations in 3D space, are fundamental. Notably, scalar properties of the atomic system, such as energy, charge, density, and physical laws, must be SE(3)-invariant, while vector properties, including force, velocity and so on, should exhibit SE(3)-equivariance.

### 3.3. EquiformerV2

Equivariant GNNs encode the node features of 3D graphs as irreps (irreducible representations) features and maintain network equivariance through message passing between nodes via equivariant operations (e.g., tensor products). Recently, notable advancements (Thomas et al., 2018; Batzner

et al., 2022; Brandstetter et al., 2021; Liao & Smidt, 2022; Liao et al., 2023) have manifested in this field. Among these, EquiformerV2 (Liao et al., 2023) stands out by scaling the architecture to higher-degree equivariant features while ensuring efficient speed.

The inference process of EquiformerV2 is represented as follows:

$$\mathbf{O}_{\texttt{EquiV2}} = \{\mathbf{O}_0, \ldots, \mathbf{O}_{l_{max}}\} = \texttt{EquiV2}(\mathbf{G}). \quad (3)$$

Here, $\mathbf{O}_l \in \mathbb{R}^{n\times c\times(2l+1)}$ represents the *type-l* output features of $n$ nodes, where $c$ is the number of channels, and $l$ ($l = 0, 1, 2, \ldots, l_{max}$) specifies the degree of equivariant features. $\mathbf{O}_0$ is aggregated using $FFN(\mathbb{R}^{n\times c\times 1} \rightarrow \mathbb{R}^{n\times d\times 1})$, followed by $Pooling(\mathbb{R}^{n\times d\times 1} \rightarrow \mathbb{R}^{d})$, yielding the overall scalar feature, where $d$ is the feature dimension. An equivariant graph attention (Liao & Smidt, 2022; Liao et al., 2023) can transform $\mathbf{O}_{\texttt{EquiV2}}$ into a new set of irrep features, where $\mathbf{O}'_1 \in \mathbb{R}^{n\times 1\times 3}$ represents desired vector properties:

$$\mathbf{O}_{\texttt{EquiAttn}} = \{\mathbf{O}'_0, \ldots, \mathbf{O}'_{l_{max}}\} = \texttt{EquiAttn}(\mathbf{O}_{\texttt{EquiV2}}, \mathbf{G}). \quad (4)$$

More details are provided in Section C for reference.

### 3.4. Rectified Flow Matching

Given two distributions $\pi_0$ and $\pi_1$, Rectified Flow Matching (Liu, 2022) aims to learn an ordinary differential equation (ODE) that maps samples $Z_0$ from $\pi_0$ to samples $Z_1$ from $\pi_1$ along a linear path that minimizes the discrepancy:

$$dZ_t = \boldsymbol{v}(Z_t, t)\, dt, \quad t \in [0, 1]. \quad (5)$$

At each time $t$, the point $Z_t$ is a linear interpolation between $Z_0$ and $Z_1$, defined as $Z_t = (1-t)Z_0 + tZ_1$. The vector field $\boldsymbol{v} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ at time $t$ directs the flow along the direction of $(Z_1 - Z_0)$, guiding $Z_t$ to follow the linear path as closely as possible. Ideally, under this vector field, $Z_t$ moves in a straight line from $Z_0$ to $Z_1$ as $t$ progresses from 0 to 1.

In practice, the vector field $\boldsymbol{v}$ is parameterized as a neural network $\boldsymbol{v}_\theta(Z_t, t)$, and the parameters $\theta$ are optimized by minimizing the following objective:

$$\int_0^1 \mathbb{E}_{Z_0\sim\pi_0, Z_1\sim\pi_1} \left\|(Z_1 - Z_0) - \boldsymbol{v}_\theta(Z_t, t)\right\|^2 dt. \quad (6)$$

Once the vector field $\boldsymbol{v}_\theta$ is trained, the ODE in Equation 5 can be solved using an ODE solver, starting from $Z_0 \sim \pi_0$ to generate samples $Z_1 \sim \pi_1$.

## 4. Method

### 4.1. SE(3)-Invariant Latent Space

As a core contribution, we introduce a novel autoencoder that directly constructs a fully SE(3)-invariant latent space while seamlessly integrating RDKit's prior knowledge, as illustrated in Figure 1 (left). The essential SE(3) equivariance of molecular conformations is preserved throughout the autoencoder's compression-reconstruction process, effectively decoupling it from the generation process on latent space.

**Encoder.** The encoder $\phi_{\textbf{\textit{Enc}}}$ is implemented as an EquiformerV2 (Liao et al., 2023), which processes the entire 3D molecule $\mathbf{G}$, composed of $\mathcal{C}$ and $\mathcal{G}$. Its purpose is to extract latent geometric features $\{\mathbf{z}_1, \mathbf{z}_2, \ldots\}$ corresponding to different low-energy conformations $\{\mathcal{C}^1, \mathcal{C}^2, \ldots\}$ of the same molecular graph $\mathcal{G}$. Different molecular graphs yield distinct collections of latent geometric features. The output features $\mathbf{O}_0$ from EquiformerV2 are aggregated into the latent variable $\mathbf{z}$, representing the geometric information of molecular conformations, via an *FFN* and *Pooling* operation. This process is formalized as follows:

$$\{\mathbf{O}_0, \ldots, \mathbf{O}_{l_{max}}\} = \phi_{\textbf{\textit{Enc}}}(\mathbf{G}) = \phi_{\textbf{\textit{Enc}}}(\mathbf{x}, \mathbf{E}, \mathcal{C}), \quad (7)$$

$$\mathbf{z} = Pooling(FFN(\mathbf{O}_0)) \in \mathbb{R}^d. \quad (8)$$

**Decoder.** The decoder $\phi_{\textbf{\textit{Dec}}}$ reconstructs the latent variable $\mathbf{z}$ into the 3D conformation $\mathcal{C}$. Inspired by DMCG (Zhu et al., 2022), a stacked reconstruction approach is employed, with the backbone consisting of multiple smaller EquiformerV2 networks $\{\varphi_{dec.1}, \varphi_{dec.2}, \ldots, \varphi_{dec.p}\}$ arranged for progressive reconstruction, where $p$ denotes the total number of cascaded networks. Initially, $\mathcal{C}_{init} \in \mathbb{R}^{n\times 3}$ is initialized as a coarse meta-conformation $\mathcal{C}_{meta}$, obtained from RDKit. For the same molecular graph $\mathcal{G}$, $\mathcal{C}_{meta}$ remains consistent. This meta-conformation is then combined with $\mathcal{G}$ to create an initial 3D molecule $\mathbf{G}_{init}$, which serves as the decoder's input. The latent variable $\mathbf{z}$ guides the reconstruction of $\mathbf{G}$ and is incorporated into the node feature encoding process of EquiformerV2 as follows (As detailed in Section C and illustrated in Figure 3(c)):

$$\mathbf{emb}_{x_i} = Embedding(x_i) + \mathbf{z}. \quad (9)$$

Subsequently, for each smaller EquiformerV2 network, The output features $\mathbf{O}'_1$ transformed by $\texttt{EquiAttn}$ are interpreted as a new conformation, denoted as $\mathcal{C}_{mid}$. The combination of $\mathcal{C}_{mid}$ and $\mathcal{G}$ forms $\mathbf{G}_{mid}$, which serves as the input for the next EquiformerV2 network. The latent variable $\mathbf{z}$ is repeatedly used as guidance for reconstruction, following the approach outlined in Equation 9. Finally, the output of the last network is taken as the reconstructed 3D molecule $\mathbf{G}_{recon}$ with its corresponding conformation $\mathcal{C}_{recon}$. The complete decoding process is formalized as follows:

$$\mathbf{G}^{(0)}_{mid} = \mathbf{G}_{init} = (\mathcal{C}_{init}, \mathcal{G}) = (\mathcal{C}_{meta}, \mathcal{G}). \quad (10)$$

$$\begin{aligned} &for\ q = 0, 1, 2, \ldots, p-1: \\ &\mathbf{O}^{(q+1)}_{\texttt{EquiV2}} = \{\mathbf{O}^{(q+1)}_0, \ldots, \mathbf{O}^{(q+1)}_{l_{max}}\} = \varphi_{dec.q+1}(\mathbf{G}^{(q)}_{mid}, \mathbf{z}); \\ &\mathcal{C}^{(q+1)}_{mid} = \mathbf{O}'_1 \ from\ \texttt{EquiAttn}(\mathbf{O}^{(q+1)}_{\texttt{EquiV2}}, \mathbf{G}^{(q)}_{mid}); \\ &\mathbf{G}^{(q+1)}_{mid} = (\mathcal{C}^{(q+1)}_{mid}, \mathcal{G}). \end{aligned} \quad (11)$$

$$(\mathcal{C}_{recon}, \mathcal{G}) = \mathbf{G}_{mid}^{(p)}. \tag{12}$$

**Training Objective.** In the pre-training phase, we adopt a loss function proposed by DMCG (Zhu et al., 2022), which is inherently strictly SE(3)-invariant for molecular conformations. We make slight modifications to the original loss function for our specific purposes. A detailed formulation of the loss function is provided in Section B.1. For simplicity, the training objective of the entire autoencoder is summarized as follows:

$$\mathcal{C}_{recon}^{*} = \rho(\sigma(\mathcal{C}_{recon})); \hat{\mathcal{C}} = Align(\mathcal{C}_{recon}^{*}, \mathcal{C}). \tag{13}$$

$$\mathcal{L}_{SE(3)\text{-}Inv}(\mathcal{C}, \mathcal{C}_{recon}) = \left\| \mathcal{C}_{recon}^{*} - \hat{\mathcal{C}} \right\|^2. \tag{14}$$

$$\mathcal{L}_{ae} = \mathbb{E}_{\mathcal{G}\sim\mathbf{D}} \left[ \mathbb{E}_{\mathcal{C}\sim\{\mathcal{C}^1,\mathcal{C}^2,\ldots\}_{\mathcal{G}}} \left[ \mathcal{L}_{SE(3)\text{-}Inv}(\mathcal{C}, \mathcal{C}_{recon}) \right] \right] + \beta \cdot \mathcal{L}_{KL}(\mathbf{z}). \tag{15}$$

In Equation 13, $\rho \in SE(3)$ represents any rigid rotation-translation operation, while $\sigma \in S$ represents any permutation operation on symmetric atoms, with $S$ denoting the set of all such operations. $Align(\mathcal{C}_a, \mathcal{C}_b)$ refers to the alignment (e.g. Kabsch (Kabsch, 1976) alignment algorithm) of $\mathcal{C}_b$ with $\mathcal{C}_a$. The alignment error $\|A\|^2$ is defined as $\sum_{i=1}^{n} \left\| \mathcal{C}_{recon\,i}^{*} - \hat{\mathcal{C}}_i \right\|^2$. The hyperparameter $\beta$ is utilized to balance the KL regularization term $\mathcal{L}_{KL}(\mathbf{z})$.

The entire construction process of the autoencoder naturally gives rise to two propositions that underpin our motivation. The formal proofs are provided in Section A.

**Proposition 1 (informal)**: We firmly believe that our latent space is SE(3)-invariant, which constitutes our main idea.

**Proposition 2 (informal)**: Furthermore, the entire autoencoder (i.e., the training objective) is also strictly SE(3)-invariant. This makes the overall structure of the autoencoder simple and elegant.

### 4.2. Latent Flow Matching

With the successful construction of an *SE(3)-invariant latent space* that effectively captures the geometric features of molecular conformations, we pioneer the application of *flow matching* (Lipman et al., 2022; Liu, 2022) within this high-quality SE(3)-invariant latent space for molecular conformation generation.

As formulated in Section 3.4, let $\pi_0$ represent the standard Gaussian distribution $N(0, I)$, and $\pi_1$ denote the true data distribution of the latent variable $\mathbf{z}$ conditioned on the molecular graph, $p(\mathbf{z}|\mathcal{G})$. Given $\mathbf{z}_0 \sim N(0, I)$ and $\mathbf{z}_1 \sim p(\mathbf{z}|\mathcal{G})$, we define the linear interpolation as $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{z}_1$, where $t \in [0, 1]$. A neural network $\boldsymbol{v}_\theta$ (illustrated in Figure 1 ( upper-right) and detailed in Section B.3) is then trained to approximate the true vector field $\boldsymbol{v}_{true} = \mathbf{z}_1 - \mathbf{z}_0$.

**Algorithm 1** Training Algorithm of CoFM

**Input:** Dataset $\mathbf{D}$, with multiple molecular graphs $\mathcal{G}$ and it's conformations set $\{\mathcal{C}^1, \mathcal{C}^2, \ldots\}$ and meta-conformation $\mathcal{C}_{meta}$ pre-calculated by RDKit.
**Initial:** Encoder $\phi_{Enc}$, Decoder $\phi_{Dec}$, Vector field $\boldsymbol{v}_\theta$.
**First Stage: Autoencoder Training**
**while** $\{\phi_{Enc}, \phi_{Dec}\}$ *not converged* **do**
  Sample $\mathcal{G} \sim \mathbf{D}$, $\mathcal{C} \sim \{\mathcal{C}^1, \mathcal{C}^2, \ldots\}$, pre-calculated $\mathcal{C}_{meta}$ of $\mathcal{G}$.
  **Encoding:**
    $\{\mathbf{O}_0, \mathbf{O}_1, \ldots, \mathbf{O}_{l_{max}}\} = \phi_{Enc}(\mathcal{G}, \mathcal{C})$.
    $\mathbf{z} = Pooling(FFN(\mathbf{O}_0))$.
  **Decoding:**
    $\mathbf{G}_{mid}^{(0)} = \mathbf{G}_{init} = (\mathcal{C}_{init}, \mathcal{G}) = (\mathbf{C}_{meta}, \mathcal{G})$.
    **for** $q = 0$ **to** $p - 1$:
      $\mathbf{O}_{\texttt{EquiV2}}^{(q+1)} = \varphi_{dec.q+1}(\mathbf{G}_{mid}^{(q)}, \mathbf{z})$;
      $\mathcal{C}_{mid}^{(q+1)} = \mathbf{O}_1'$ *from* $\texttt{EquiAttn}(\mathbf{O}_{\texttt{EquiV2}}^{(q+1)}, \mathbf{G}_{mid}^{(q)})$;
      $\mathbf{G}_{mid}^{(q+1)} = (\mathcal{C}_{mid}^{(q+1)}, \mathcal{G})$.
    **end for**
    $(\mathcal{C}_{recon}, \mathcal{G}) = \mathbf{G}_{mid}^{(p)}$.
  $\mathcal{L}_{ae} = \mathcal{L}_{SE(3)\text{-}Inv}(\mathcal{C}, \mathcal{C}_{recon}) + \beta \mathcal{L}_{KL}(\mathbf{z})$.
  Update $\phi_{Enc}$, $\phi_{Dec}$ using $\mathcal{L}_{ae}$.
**end while**
**Second Stage: Latent Flow Matching Training**
**while** $\boldsymbol{v}_\theta$ *not converged* **do**
  Sample $\mathcal{G} \sim \mathbf{D}$, $\mathcal{C} \sim \{\mathcal{C}^1, \mathcal{C}^2, \ldots\}$.
  $\mathbf{z}_1 = \phi_{Enc}(\mathcal{G}, \mathcal{C})$, Sample $\mathbf{z}_0 \sim N(0, I), t \sim \mathcal{U}(0, 1)$.
  $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{z}_1, \boldsymbol{v}_{true} = \mathbf{z}_1 - \mathbf{z}_0$.
  $\mathcal{L}_{fm} = \|\boldsymbol{v}_{true} - \boldsymbol{v}_\theta(\mathbf{z}_t, t, \mathcal{G})\|^2$.
  Update $\boldsymbol{v}_\theta$ using $\mathcal{L}_{fm}$.
**end while**
**Return:** $\phi_{Enc}$, $\phi_{Dec}$, $\boldsymbol{v}_\theta$.

**Algorithm 2** Sampling Algorithm of CoFM

**Input:** Molecular graph $\mathcal{G}$, $\mathbf{C}_{meta}$ of $\mathcal{G}$.
**Initial:** Pre-trained decoder $\phi_{Dec}$, Vector field $\boldsymbol{v}_\theta$, Time steps $N$, $\Delta t = 1/N$, $t = 0$.
**Sampling:**
  **Sample** $\mathbf{z}_t \sim \mathcal{N}(0, \sigma_0^2 I)$.
  $\mathbf{z}_t = \mathbf{z}_t + \boldsymbol{v}_\theta(\mathbf{z}_t, t, \mathcal{G}) \cdot \Delta t$, **for** $i = 0$ **to** $N - 1$.
  $\mathbf{z}_1^{\text{gen}} = \mathbf{z}_t$.
**Decoding:**
  $\mathcal{C}_{gen} = \phi_{Dec}(\mathcal{G}, \mathbf{z}_1^{\text{gen}}, \mathbf{C}_{meta})$.
**Return:** $(\mathcal{C}_{gen}, \mathcal{G})$.

The final training objective is formalized as:

$$\mathbb{E}_{\mathbf{z}_0\sim\pi_0, \mathbf{z}_1\sim\pi_1, t\in\mathcal{U}(0,1)} \left\| \boldsymbol{v}_{true} - \boldsymbol{v}_\theta(\mathbf{z}_t, t, \mathcal{G}) \right\|^2. \tag{16}$$

After training, we sample $\mathbf{z}_0 \sim \mathcal{N}(0, \sigma_0^2 I)$. Notably, during sampling, we observed that flexibly adjusting the standard

deviation $\sigma_0$ of the Gaussian prior leads to a more effective balance in performance. The sample is then iteratively updated using a simple Euler method:

$$\mathbf{z}_t = \mathbf{z}_0 + \boldsymbol{v}_\theta(\mathbf{z}_t, t, \mathcal{G}) \cdot \Delta t. \tag{17}$$

This approach moves $\mathbf{z}_t$ along the direction of the learned vector field $\boldsymbol{v}_\theta(\mathbf{z}_t, t, \mathcal{G})$ toward the true data distribution $p(\mathbf{z}|\mathcal{G})$, thereby generating a new latent variable $\mathbf{z}_1^{\text{gen}}$.

### 4.3. CoFM Overview

As shown in Figure 1 and detailed in Section 4.1 and Section 4.2, the core concept of **CoFM** lies in the design of an elegant and easily implementable autoencoder. This autoencoder *induces a fully SE(3)-invariant latent space* while seamlessly integrating RDKit's prior knowledge, effectively capturing the geometric features of molecular conformations, which represents a key advancement of this work. Leveraging this SE(3)-invariant latent space, the latest *rectified flow matching* technique is applied to the molecular conformation generation task in non-3D coordinate space for the first time, yielding significant improvements in both generation efficiency and outcome reliability. The complete training and generation processes of the algorithm are outlined in Algorithm 1 and Algorithm 2, offering a comprehensive understanding of the approach.

## 5. Experiments

### 5.1. Datasets and Baselines

**Datasets.** Building on previous benchmark studies, we use the Small-scale GEOM (Axelrod & Gomez-Bombarelli, 2020) dataset to evaluate molecular conformation generation. The Small-scale QM9 dataset includes small organic molecules with up to 9 heavy atoms, while Small-scale Drugs consists of larger drug-like molecules with up to 91 heavy atoms. To ensure fair comparisons, we follow the ConfGF (Shi et al., 2021) and GeoDiff (Xu et al., 2022). Both datasets have training sets of 40,000 molecules, each with 5 low-energy conformations. The test sets contain 200 molecules, with 22,408 conformations in QM9 and 14,324 in Drugs. The remaining molecules form the validation set, each having 5 conformations consistent with the training set. Furthermore, we follow ETFlow (Hassan et al., 2024) and compare our method with the latest approaches on the large-scale QM9 data set.

It is important to note that our method relies on the initial conformation $\mathbf{C}_{meta}$ generated by RDKit (Landrum, 2006), which serves as the basis for subsequent optimization and learning. RDKit is generally robust in producing reasonable initial conformations, and in most cases, the chosen conformation provides a good starting point. However, poor or atypical $\mathbf{C}_{meta}$ (e.g. with unrealistic geometry) can negatively affect the model's performance, leading to slower convergence or inaccurate reconstruction. Nevertheless, due to RDKit's robustness, the risk of such issues is minimal in practice. In our implementation, we use the ETKDG algorithm in RDKit during data pre-processing to generate $\mathbf{C}_{meta}$. While multiple conformations could be generated, we run it once to ensure consistency for the same molecular graph.

**Baselines.** We compare our method against several ML-based approaches that have demonstrated excellent performance, including CGCF (Xu et al., 2021a), ConfGF (Shi et al., 2021), ConfVAE (Xu et al., 2021b), GeoMol (Ganea et al., 2021), GeoDiff (Xu et al., 2022), DMCG (Zhu et al., 2022), SDE-Gen (Zhang et al., 2023), EC-Conf (Fan et al., 2023), TorDiff (Jing et al., 2022), MCF (Wang et al., 2023) and ETFlow (Hassan et al., 2024).

### 5.2. Results and Analysis

The diversity and quality of molecular conformation generation are assessed using four indicators, as detailed in Section B.2, with the key model setup information provided in Section B.6.

**Coverage and Matching.** Table 1 presents the Recall and Precision metrics for the Small-scale QM9 dataset. CoFM outperforms all other methods in key Precision metrics, particularly after just 2 sampling steps. For the mean value of COV-P, CoFM achieves an impressive 93.76%, exceeding the second-best method, DMCG, by 6.5%. Similarly, for the mean value of MAT-P, CoFM achieves the lowest value of 0.2168 Å after 2 sampling steps, representing a 24.51% improvement over DMCG.These results highlight CoFM 's remarkable capability in generating high-quality molecular conformations with fewer sampling steps compared to diffusion-based models. Moreover, as the number of sampling steps increases, CoFM maintains competitive Recall performance compared to the best methods. If greater diversity is required for practical applications, increasing the number of sampling steps for CoFM enhances diversity, as reflected in improved Recall metrics. While the gains in diversity may not surpass those of other methods, CoFM still achieves the second-best performance in COV-R and the best performance in MAT-R with 25 sampling steps. This demonstrates CoFM 's flexibility in balancing diversity and quality. Table 2 presents the results on the Large-scale QM9 dataset. CoFM achieves the highest COV-P and the second-best MAT-P using only 5 sampling steps. Notably, it significantly outperforms TorDiff, another method that incorporates RDKit-based physicochemical priors. The results for the Small-scale Drugs dataset are shown in Table 4 in Section B.4. CoFM achieves the best performance in Precision metrics, particularly after 5 sampling steps. Although its Recall metrics are less competitive compared to

*Table 1.* Results on the Small-scall QM9 dataset.

| Methods | Steps | COV-R (%) ↑ | | MAT-R (Å) ↓ | | COV-P (%) ↑ | | MAT-P (Å) ↓ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| RDKit | 1 | 83.26 | 90.78 | 0.3447 | 0.2935 | - | - | - | - |
| GraphDG | 1 | 73.33 | 84.21 | 0.4245 | 0.3973 | 43.90 | 35.33 | 0.5809 | 0.5823 |
| ConfVAE | 1 | 77.84 | 88.20 | 0.4154 | 0.3739 | 38.02 | 34.67 | 0.6215 | 0.6091 |
| Geomol | 1 | 71.26 | 72.00 | 0.3731 | 0.3731 | - | - | - | - |
| DMCG | 1 | **96.23** | **99.26** | **0.2083** | **0.2014** | **87.26** | **91.00** | **0.2872** | **0.2926** |
| ***3D Space Diffusion*** | | | | | | | | | |
| CGCF | 1000 | 78.05 | 82.48 | 0.4219 | 0.3900 | 36.49 | 33.57 | 0.6615 | 0.6427 |
| ConfGF | 5000 | 88.49 | 94.31 | 0.2673 | 0.2685 | 46.43 | 43.41 | 0.5224 | 0.5124 |
| SDEGen[a] | 1500 | 81.53 | 85.99 | 0.3568 | 0.3612 | 48.37 | 46.63 | 0.5662 | 0.5483 |
| GeoDiff-A[b] | 5000 | **90.54** | **94.61** | 0.2104 | 0.2021 | 52.35 | 50.10 | 0.4539 | 0.4399 |
| GeoDiff-C[c] | 5000 | 90.07 | 93.39 | **0.2090** | **0.1988** | 52.79 | 50.29 | 0.4448 | 0.4267 |
| EcConf | 2 | 75.89 | 79.71 | 0.4087 | 0.4016 | 79.85 | 82.85 | 0.4128 | 0.4176 |
| EcConf | 5 | 82.95 | 88.06 | 0.3475 | 0.3440 | 83.24 | 87.66 | 0.3777 | 0.3732 |
| EcConf | 25 | 82.35 | 86.54 | 0.3223 | 0.3196 | **86.30** | **90.88** | 0.3368 | 0.3356 |
| ***SE(3)-Inv Latent Flow Matching (Ours)*** | | | | | | | | | |
| **CoFM** | 2 | 90.20 | 94.55 | 0.2167 | 0.2018 | **93.76** | **97.62** | **0.2168** | **0.2134** |
| **CoFM** | 5 | 92.25 | 96.64 | **0.2001** | **0.1819** | 92.50 | 96.67 | 0.2254 | 0.2216 |
| **CoFM** | 25 | **93.32** | 97.78 | **0.2001** | 0.1932 | 90.92 | 94.90 | 0.2399 | 0.2336 |
| **CoFM** | 50 | 93.30 | **97.94** | 0.2008 | 0.1920 | 90.70 | 94.55 | 0.2424 | 0.2550 |

*Table 2.* Results on the Large-scale QM9 dataset.

| Methods | Steps | COV-R (%) ↑ | | MAT-R (Å) ↓ | | COV-P (%) ↑ | | MAT-P (Å) ↓ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| GeoMol | 1 | 91.50 | **100.00** | 0.225 | 0.193 | 87.60 | **100.00** | 0.270 | 0.241 |
| ***3D Space Diffusion*** | | | | | | | | | |
| CGCF | 1000 | 69.47 | 96.15 | 0.425 | 0.374 | 38.20 | 33.33 | 0.711 | 0.695 |
| GeoDiff | 5000 | 76.50 | **100.00** | 0.297 | 0.229 | 50.00 | 33.50 | 1.524 | 0.510 |
| TorDiff | 20 | 92.80 | **100.00** | 0.178 | 0.147 | 92.70 | **100.00** | 0.221 | 0.195 |
| MCF | 1000 | 95.00 | **100.00** | 0.103 | 0.044 | 93.70 | **100.00** | 0.119 | 0.055 |
| ET-Flow | 50 | **96.47** | **100.00** | **0.073** | 0.047 | 94.05 | **100.00** | **0.098** | **0.039** |
| ET-Flow - SO(3) | 50 | 95.98 | **100.00** | 0.076 | **0.030** | 92.10 | **100.00** | 0.110 | 0.047 |
| ***SE(3)-Inv Latent Flow Matching (Ours)*** | | | | | | | | | |
| **CoFM** | 5 | 95.41 | **100.00** | 0.105 | 0.060 | **95.50** | **100.00** | 0.109 | 0.066 |
| **CoFM** | 50 | 95.46 | **100.00** | 0.101 | 0.059 | 95.13 | **100.00** | 0.111 | 0.064 |

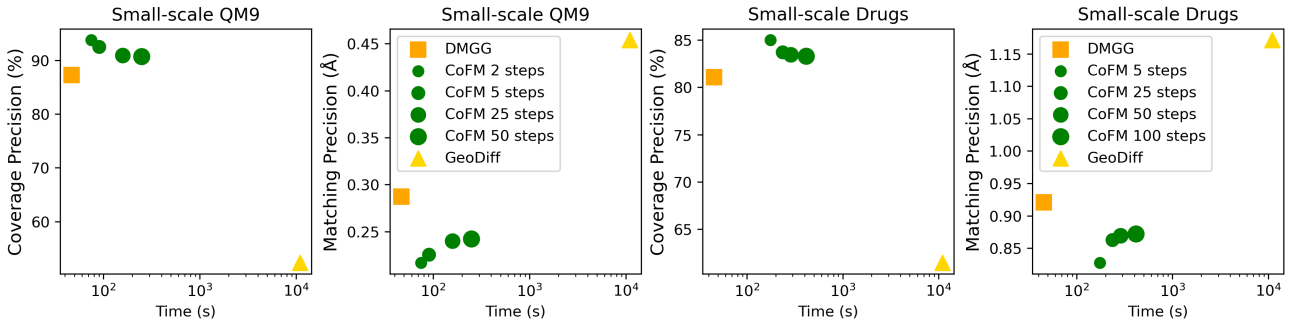



*Figure 2.* **The performance efficiency of various methods on the Small-scale QM9 and Drugs test sets.**. The x-axis represents the overall inference time (in seconds), while the y-axis represents the COV-P, and MAT-P metrics respectively.

other methods, CoFM still attains the second-best MAT-R after 100 sampling steps. These findings highlight CoFM 's capability to generate high-quality molecular conformations for larger molecules.

*Table 3.* MAE of predicted ensemble properties (eV).

| Method | $\overline{E}$ | $E_{\text{min}}$ | $\overline{\Delta\epsilon}$ | $\Delta\epsilon_{\text{min}}$ | $\Delta\epsilon_{\text{max}}$ |
|---|---|---|---|---|---|
| RDKit | 0.9233 | 0.6585 | 0.3698 | 0.8021 | 0.2359 |
| GraphDG | 9.1027 | 0.8882 | 1.7973 | 4.1743 | 0.4776 |
| CGCF | 28.9661 | 2.8410 | 2.8356 | 10.6361 | 0.5954 |
| ConfVAE | 8.2080 | 0.6100 | 1.6080 | 3.9111 | 0.2429 |
| ConfGF | 2.7886 | 0.1765 | 0.4688 | 2.1843 | **0.1433** |
| GeoDiff | 0.2597 | 0.1551 | 0.3091 | 0.7033 | 0.1909 |
| DMCG | 0.4324 | **0.1364** | 0.2057 | 1.3229 | 0.1509 |
| **CoFM** | **0.0631** | 0.6478 | **0.0796** | **0.1508** | 0.1603 |

**Properties Error.** To evaluate the *molecular ensemble properties* of the generated conformations as an indicator of their quality, we follow methodologies established in prior studies (Xu et al., 2021a; Shi et al., 2021; Zhu et al., 2022). Specifically, thirty molecules are randomly selected from the Small-scale QM9 test set, and 50 conformations are generated for each molecule. Using the quantum chemistry software Psi4 (Smith et al., 2020), we calculate each conformer's energy ($E$) and HOMO-LUMO gap ($\epsilon$). Key ensemble metrics are then assessed, including the mean absolute error (MAE) of the average energy ($\overline{E}$), the lowest energy ($E_{\text{min}}$), the average gap ($\overline{\Delta\epsilon}$), the minimum gap ($\Delta\epsilon_{\text{min}}$), and the maximum gap ($\Delta\epsilon_{\text{max}}$), relative to those of the ground truth conformations. As shown in Table 3, CoFM achieves the lowest errors in $\overline{E}$, $\overline{\Delta\epsilon}$, and $\Delta\epsilon_{\text{min}}$ compared to the ground truth values. This highlights its remarkable ability to accurately capture both the average energy and the intricate electronic properties of the molecular ensemble. These results underscore the model's superior performance in generating high-quality conformations with consistent and reliable ensemble energy properties.

**Inference Efficiency.** To assess our method's ability to generate high-quality molecular conformations with realistic energy properties while ensuring superior inference efficiency, we measured the time taken by different methods to complete inference on the entire Small-scale QM9 and Drugs test sets, using the same batch size. For our proposed CoFM, we evaluated inference times with 2, 5, 25, and 50 sampling steps for QM9 (and 5, 25, 50, and 200 steps for Drugs), comparing the results with those of the DMCG (Zhu et al., 2022) method and the representative 3D spatial diffusion model algorithm GeoDiff (Xu et al., 2022). The results, shown in Figure 2, demonstrate that DMCG achieves the fastest inference time. However, CoFM remains highly competitive, offering significant improvements in COV-P and MAT-P. In contrast, GeoDiff requires considerably longer inference times and shows lower performance. These findings emphasize CoFM's ability to efficiently generate high-quality molecular conformations with realistic energy properties, while maintaining competitive inference speeds compared to traditional 3D spatial diffusion models.

**Visualization.** To provide a more intuitive visualization, several representative examples are presented. Figure 4 showcases cases from the Small-scale QM9 test set, while Figure 5 highlights examples from the Small-scale Drugs test set. As shown in the figures, CoFM generates numerous conformations that closely resemble the reference conformations.

## 6. Conclusion

We present **CoFM**, a novel method for molecular conformation generation that combines innovation, efficiency, and accuracy. At its core is an advanced autoencoder framework that constructs a fully SE(3)-invariant latent space, seamlessly integrating RDKit's robust prior knowledge while confining SE(3) equivariance constraints to the autoencoder's reconstruction process. This design enables the model to more effectively capture the intricate geometric feature distributions of conformations encoded in the latent vector. By introducing the flow matching technique into the SE(3)-invariant latent space for the first time, **CoFM** achieves significant advancements in both generation efficiency and the energy reliability of the resulting conformations. Our method consistently delivers high-quality results across diverse and complex molecular datasets, demonstrating its ability to balance diversity and precision while maintaining competitive inference speed.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References


Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Axelrod, S. and Gomez-Bombarelli, R. Geom: Energy-annotated molecular conformations for property prediction and molecular generation. *arXiv preprint arXiv:2006.05531*, 2020.

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. *Nature communications*, 13(1):2453, 2022.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. Geometric and physical quantities improve e (3) equivariant message passing. *arXiv preprint arXiv:2110.02905*, 2021.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Fan, Z., Yang, Y., Xu, M., and Chen, H. Ec-conf: A ultra-fast diffusion model for molecular conformation generation with equivariant consistency. *arXiv preprint arXiv:2308.00237*, 2023.

Ganea, O., Pattanaik, L., Coley, C., Barzilay, R., Jensen, K., Green, W., and Jaakkola, T. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 34: 13757–13769, 2021.

Halgren, T. A. Merck molecular force field. v. extension of mmff94 using experimental data, additional computational data, and empirical rules. *Journal of Computational Chemistry*, 17(5-6):616–641, 1996.

Hassan, M., Shenoy, N., Lee, J., Stark, H., Thaler, S., and Beaini, D. Et-flow: Equivariant flow-matching for molecular conformer generation. *arXiv preprint arXiv:2410.22388*, 2024.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Howard, A. G. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35: 24240–24253, 2022.

Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Landrum, G. Rdkit: Open-source cheminformatics. http://www.rdkit.org, 2006.

Liao, Y.-L. and Smidt, T. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

Liao, Y.-L., Wood, B., Das, A., and Smidt, T. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

Liberti, L., Lavor, C., Maculan, N., and Mucherino, A. Euclidean distance geometry and applications. *SIAM review*, 56(1):3–69, 2014.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, Q. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2018.

Parr, R. G., Gadre, S. R., and Bartolotti, L. J. Local density functional theory of atoms and molecules. *Proceedings of the National Academy of Sciences*, 76(6):2522–2526, 1979.

Passaro, S. and Zitnick, C. L. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. In *International Conference on Machine Learning*, pp. 27420–27438. PMLR, 2023.

Rappé, A. K., Casewit, C. J., Colwell, K., Goddard III, W. A., and Skiff, W. M. 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.

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

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation. In *International conference on machine learning*, pp. 9558–9568. PMLR, 2021.

Simm, G. N. and Hernández-Lobato, J. M. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

Smith, D. G. A., Burns, L., Simmonett, A., Parrish, R., Schieber, M. C., Galvelis, R., Kraus, P., Kruse, H., Remigio, R. D., Alenaizan, A., James, A. M., Lehtola, S., Misiewicz, J. P., et al. Psi4 1.4: Open-source software for high-throughput quantum chemistry. *The Journal of chemical physics*, 2020.

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

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Thölke, P. and De Fabritiis, G. Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.

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. *arXiv preprint arXiv:1802.08219*, 2018.

Wang, Y., Elhag, A. A., Jaitly, N., Susskind, J. M., and Bautista, M. A. Swallowing the bitter pill: Simplified scalable conformer generation. *arXiv preprint arXiv:2311.17932*, 2023.

Xu, M., Luo, S., Bengio, Y., Peng, J., and Tang, J. Learning neural generative dynamics for molecular conformation generation. *arXiv preprint arXiv:2102.10240*, 2021a.

Xu, M., Wang, W., Luo, S., Shi, C., Bengio, Y., Gomez-Bombarelli, R., and Tang, J. An end-to-end framework for molecular conformation generation via bilevel programming. In *International Conference on Machine Learning*, pp. 11537–11547. PMLR, 2021b.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Zhang, H., Li, S., Zhang, J., Wang, Z., Wang, J., Jiang, D., Bian, Z., Zhang, Y., Deng, Y., Song, J., et al. Sdegen: learning to evolve molecular conformations from thermodynamic noise for conformation generation. *Chemical Science*, 14(6):1557–1568, 2023.

Zhu, J., Xia, Y., Liu, C., Wu, L., Xie, S., Wang, Y., Wang, T., Qin, T., Zhou, W., Li, H., et al. Direct molecular conformation generation. *arXiv preprint arXiv:2202.01356*, 2022.

## A. Proofs of Propositions

**Proposition A.1 (SE(3)-invariance of the latent vector).** *Let $\mathbf{G} = \{\mathbf{x}, \mathbf{E}, \mathcal{C}\}$ be a 3D molecular graph with $n$ atoms, where $\mathcal{C} \in \mathbb{R}^{n\times 3}$ denotes the atomic coordinates. Let $\rho\colon \mathbb{R}^3 \to \mathbb{R}^3$ be a rigid-body transformation in the special Euclidean group $\mathrm{SE}(3)$, i.e., $\rho(\mathbf{r}) = \mathbf{R}\mathbf{r} + \mathbf{t}$ for some $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$. We define the transformed conformation as $\rho(\mathcal{C}) \in \mathbb{R}^{n\times 3}$, with $\rho(\mathcal{C})_i = \rho(\mathcal{C}_i)$ applied row-wise. Let $\mathcal{G} = \{\mathbf{x}, \mathbf{E}\}$ denote the molecular topology. Let $\phi_{Enc} : (\mathcal{C}, \mathcal{G}) \mapsto \mathbf{z} \in \mathbb{R}^d$ be an encoder composed of:*

*1. an* $\mathrm{SE}(3)$-equivariant *graph network $\Phi$ (e.g., EquiFormerV2 (Liao et al., 2023)) producing per-node scalar features $\mathbf{O}_0 = (\mathbf{o}_1, \ldots, \mathbf{o}_n) \in \mathbb{R}^{n\times d}$,*

*2. a feed-forward network FFN$\colon \mathbb{R}^d \to \mathbb{R}^d$ applied to each node independently,*

*3. a permutation-invariant pooling operator Pooling, e.g., mean or sum pooling.*

*Then the latent vector $\mathbf{z}$ is invariant under any rigid-body transformation $\rho \in \mathrm{SE}(3)$, i.e.,*

$$\phi_{Enc}(\mathcal{C}, \mathcal{G}) = \phi_{Enc}(\rho(\mathcal{C}), \mathcal{G}).$$

*Proof.* **Step 1 (SE(3)-invariant features).** The equivariant network $\Phi$ outputs scalar features for each node, which are invariant under $\rho \in \mathrm{SE}(3)$:

$$\mathbf{O}_0 = \Phi(\mathcal{C}, \mathcal{G}) = \Phi(\rho(\mathcal{C}), \mathcal{G}) \in \mathbb{R}^{n\times d}.$$

**Step 2 (Feed-forward projection).** Applying *FFN* independently to each row preserves the equality:

$$\mathit{FFN}(\mathbf{O}_0) = \mathit{FFN}(\Phi(\mathcal{C}, \mathcal{G})) = \mathit{FFN}(\Phi(\rho(\mathcal{C}), \mathcal{G})).$$

**Step 3 (Pooling).** Since *Pooling* is permutation-invariant and coordinate-independent:

$$\mathbf{z} = \mathit{Pooling}(\mathit{FFN}(\mathbf{O}_0)) = \mathit{Pooling}(\mathit{FFN}(\mathbf{O}_0^\rho)) = \phi_{Enc}(\rho(\mathcal{C}), \mathcal{G}).$$

**Conclusion.** Thus, we have

$$\phi_{Enc}(\mathcal{C}, \mathcal{G}) = \phi_{Enc}(\rho(\mathcal{C}), \mathcal{G}), \quad \forall \rho \in \mathrm{SE}(3).$$

□

**Proposition A.2 (SE(3)-invariance of the full autoencoder).** *Let $\mathbf{G} = \{\mathbf{x}, \mathbf{E}, \mathcal{C}\}$ be a 3D molecular graph with conformation $\mathcal{C} \in \mathbb{R}^{n\times 3}$, and let $\rho \in \mathrm{SE}(3)$ be any rigid-body transformation applied row-wise to $\mathcal{C}$. Let $\phi_{Enc}$ be the encoder producing latent vector $\mathbf{z}$, and $\phi_{Dec}$ the decoder reconstructing a conformation from $\mathbf{z}$. Then the entire objective of the autoencoder is SE(3)-invariant:*

$$\mathcal{L}_{ae}(\mathcal{C}, \widehat{\mathcal{C}}) = \mathcal{L}_{ae}(\rho(\mathcal{C}), \widehat{\mathcal{C}}_\rho),$$

*where*

$$\widehat{\mathcal{C}} = \phi_{Dec}(\mathcal{G}, \mathcal{C}_{meta}, \phi_{Enc}(\mathcal{C})), \quad \widehat{\mathcal{C}}_\rho = \phi_{Dec}(\mathcal{G}, \mathcal{C}_{meta}, \phi_{Enc}(\rho(\mathcal{C}))).$$

*Proof.* **Encoder invariance.** From Proposition A.1, we have

$$\phi_{Enc}(\rho(\mathcal{C}), \mathcal{G}) = \phi_{Enc}(\mathcal{C}, \mathcal{G}),$$

i.e., the latent vector $\mathbf{z}$ is SE(3)-invariant.

**Decoder equivariance.** The decoder $\phi_{Dec}$ is composed of multiple SE(3)-*equivariant* layers (e.g., EquiFormerV2 (Liao et al., 2023) blocks), each of which takes node-wise features (augmented with $\mathbf{z}$) and outputs coordinate predictions. Since the input to the decoder includes the SE(3)-invariant latent vector and the fixed topology $\mathcal{G}$, and each block preserves equivariance, the output satisfies:

$$\widehat{\mathcal{C}}_\rho = \rho(\widehat{\mathcal{C}}),$$

i.e., the reconstructed conformation transforms consistently with the input.

**Loss invariance.** Let $\mathcal{L}_{ae}(\cdot,\cdot)$ denote the reconstruction loss, specified in Section B.1. Since the decoder is SE(3)-equivariant and the loss function is defined in an SE(3)-invariant way, we have:

$$\mathcal{L}_{ae}(\rho(\mathcal{C}),\rho(\widehat{\mathcal{C}})) = \mathcal{L}_{ae}(\mathcal{C},\widehat{\mathcal{C}}).$$

**Conclusion.** Combining the encoder invariance, decoder equivariance, and invariant loss yields the desired result:

$$\mathcal{L}_{ae}(\rho(\mathcal{C}),\widehat{\mathcal{C}}_\rho) = \mathcal{L}_{ae}(\mathcal{C},\widehat{\mathcal{C}}).$$

□

## B. More Details and Results of CoFM

### B.1. SE(3)-Invariant Loss Function

For our autoencoder training, we employ a fully SE(3)-invariant loss function adapted from DMCG (Zhu et al., 2022). Specifically, let $\rho$ represents any rigid rotation-translation operation, while $\sigma$ represents any permutation operation on symmetric atoms, with $S$ denoting the set of all such operations. The primary component is the alignment loss, $\mathcal{L}_{\textit{SE(3)-Inv}}(\mathcal{C},\mathcal{C}_{\textit{recon}})$, as defined in Equation 14.

Furthermore, following DMCG (Zhu et al., 2022), an angle loss and a bond loss are introduced to enhance the training process. Let $\mathcal{C}$ represent the ground truth conformation and $\mathcal{C}_{\textit{recon}}$ denote the reconstructed conformation. Define $E$ as the set of all bond connections and $E_2$ as the set of triplets $(i,j,k)$ satisfying $(i,j)\in E$, $(i,k)\in E$, and $k\neq j$.

The coordinates of the $i$-th node are given by $\mathcal{C}_i$, and the vector between the $i$-th and $j$-th nodes is defined as:

$$\vec{r}_{ij} = \mathcal{C}_j - \mathcal{C}_i. \tag{18}$$

Additionally, the distance matrix $\mathcal{D}\in\mathbb{R}^{n\times n}$ is introduced, where $\mathcal{D}_{ij}$ represents the distance between nodes $i$ and $j$, formally defined as:

$$\mathcal{D}_{ij} = \|\mathcal{C}_j - \mathcal{C}_i\|. \tag{19}$$

Similarly, for the reconstructed conformation, we define $\mathcal{C}_i^{\textit{recon}}$ as the coordinates of the $i$-th node, $\vec{r}_{ij}^{\,\textit{recon}}$ as the vector between nodes $i$ and $j$, and $\mathcal{D}^{\textit{recon}}\in\mathbb{R}^{n\times n}$ as the corresponding distance matrix, where:

$$\mathcal{D}_{ij}^{\textit{recon}} = \|\mathcal{C}_j^{\textit{recon}} - \mathcal{C}_i^{\textit{recon}}\|. \tag{20}$$

The angle loss and bond loss are then formulated as follows:

$$\mathcal{L}_{\textit{angle}} = \frac{1}{|E_2|}\sum_{(i,j,k)\in E_2}\left\|cosine(\vec{r}_{ij},\vec{r}_{ik}) - cosine(\vec{r}_{ij}^{\,\textit{recon}},\vec{r}_{ik}^{\,\textit{recon}})\right\|_F^2 \tag{21}$$

$$\mathcal{L}_{\textit{bond}} = \frac{1}{|E|}\sum_{(i,j)\in E}\left(\mathcal{D}_{ij} - \mathcal{D}_{ij}^{\textit{recon}}\right)^2 \tag{22}$$

Additionally, we introduce a distance loss $\mathcal{L}_{\textit{distance}}$, defined as the MAE error between $\mathcal{D}$ and $\mathcal{D}_{\textit{recon}}$, which optimizes the distances between non-bonded nodes. This loss has the potential to facilitate convergence.

Overall, given a ground truth conformation $\mathcal{C}$ and a reconstructed conformation $\mathcal{C}\textit{recon}$, let $\lambda_1$, $\lambda_2$, $\lambda_3$, and $\lambda_4$ denote the hyperparameters controlling the weights of each loss component. The reconstruction loss is then defined as:

$$\mathcal{L}_{\textit{recon}} = \lambda_1\cdot\mathcal{L}_{\textit{SE(3)-Inv}}(\mathcal{C},\mathcal{C}_{\textit{recon}}) + \lambda_2\cdot\mathcal{L}_{\textit{angle}} + \lambda_3\cdot\mathcal{L}_{\textit{bond}} + \lambda_4\cdot\mathcal{L}_{\textit{distance}} \tag{23}$$

In practical implementation, we employ a cascade decoder with outputs such as $\mathcal{C}_{\textit{mid}}^{(1)},\mathcal{C}_{\textit{mid}}^{(2)},\ldots,\mathcal{C}_{\textit{recon}}$. For each intermediate output and the final reconstructed conformation, we compute the respective reconstruction losses. Additionally, a KL divergence loss is applied as a regularization term, where $\mathbf{z}$ represents the latent representation produced by the encoder. Let $\alpha$ and $\beta$ be hyperparameters to balance the reconstruction loss and KL regularization. The overall loss for our autoencoder is formulated as follows:

$$\mathcal{L}_{\textit{ae}} = \alpha\cdot[\mathcal{L}_{\textit{recon}}(\mathcal{C},\mathcal{C}_{\textit{mid}}^{(1)}) + \mathcal{L}_{\textit{recon}}(\mathcal{C},\mathcal{C}_{\textit{mid}}^{(2)}) + \ldots + \mathcal{L}_{\textit{recon}}(\mathcal{C},\mathcal{C}_{\textit{recon}})] + \beta\cdot\mathcal{L}_{\textit{KL}}(\mathbf{z}) \tag{24}$$

**Proposition B.1 (SE(3)-invariance of the training objective).** *Let $\mathbf{G} = \{\mathbf{x}, \mathbf{E}, \mathcal{C}\}$ be a 3D molecular graph with coordinates $\mathcal{C} \in \mathbb{R}^{n\times 3}$, and let $\rho \in \mathrm{SE}(3)$ be any rigid-body transformation applied row-wise to $\mathcal{C}$. Let $\phi_{Enc}(\mathcal{C}, \mathcal{G})$ be the encoder producing a latent vector $\mathbf{z} \in \mathbb{R}^d$, and let the reconstructed conformation be given by*

$$\widehat{\mathcal{C}} = \phi_{Dec}(\mathcal{G}, \mathcal{C}_{meta}, \phi_{Enc}(\mathcal{C})).$$

*Then the total training loss*

$$\mathcal{L}_{ae} = \mathcal{L}_{KL} + \mathcal{L}_{recon} + \mathcal{L}_{angle} + \mathcal{L}_{bond} + \mathcal{L}_{distance}$$

*is strictly SE(3)-invariant, i.e.,*

$$\mathcal{L}_{ae}(\mathcal{C}, \widehat{\mathcal{C}}) = \mathcal{L}_{ae}(\rho(\mathcal{C}), \rho(\widehat{\mathcal{C}})).$$

*Proof.* **Step 1 (KL term).** The latent vector $\mathbf{z} = \phi_{Enc}(\mathcal{C}, \mathcal{G})$ is SE(3)-invariant by Proposition A.1, and hence the KL divergence loss $\mathcal{L}_{KL}(\mathbf{z})$ is also SE(3)-invariant.

**Step 2 (Reconstruction loss).** The reconstruction loss $\mathcal{L}_{recon}$ is defined as a comparison between the predicted conformation $\widehat{\mathcal{C}}$ and the reference $\mathcal{C}$, using an SE(3)-invariant metric such as pairwise distances or Kabsch-aligned coordinates. Thus,

$$\mathcal{L}_{recon}(\mathcal{C}, \widehat{\mathcal{C}}) = \mathcal{L}_{recon}(\rho(\mathcal{C}), \rho(\widehat{\mathcal{C}})).$$

**Step 3 (Angle, bond, and distance losses).** Each of the remaining terms:

$$\mathcal{L}_{angle}, \quad \mathcal{L}_{bond}, \quad \mathcal{L}_{distance}$$

is computed from internal geometric quantities such as interatomic angles, bond lengths, or pairwise distances, which are invariant under SE(3) transformations. Therefore, each satisfies:

$$\mathcal{L}_{*}(\mathcal{C}, \widehat{\mathcal{C}}) = \mathcal{L}_{*}(\rho(\mathcal{C}), \rho(\widehat{\mathcal{C}})), \quad * \in \{\text{angle, bond, distance}\}.$$

**Conclusion.** Since each term in $\mathcal{L}_{\text{ae}}$ is SE(3)-invariant, the total loss is also SE(3)-invariant:

$$\mathcal{L}_{ae}(\mathcal{C}, \widehat{\mathcal{C}}) = \mathcal{L}_{ae}(\rho(\mathcal{C}), \rho(\widehat{\mathcal{C}})), \quad \forall \rho \in \mathrm{SE}(3).$$

□

### B.2. Evaluation Metrics

**RMSD.** In the field of bioinformatics, RMSD is commonly utilized to quantitatively assess the dissimilarity between two 3D structures. Its calculation involves normalizing the Frobenius norm of the aligned coordinate matrices after applying the Kabsch algorithm (Kabsch, 1976), as described by equation 25:

$$\mathrm{RMSD}(\mathcal{C}, \tilde{\mathcal{C}}') = \sqrt{\frac{1}{n}\sum_{i=1}^{N} \left\| \mathcal{C}_i - \tilde{\mathcal{C}}'_i \right\|^2}. \tag{25}$$

where $\tilde{\mathcal{C}}'$ represents the generated conformation after alignment using the Kabsch algorithm, and $n$ represents the number of atoms. $\mathcal{C}_i$ and $\tilde{\mathcal{C}}'_i$ represent the coordinate vectors of the i-th atom in $\mathcal{C}$ and $\tilde{\mathcal{C}}'$, respectively, in 3D space.

**Coverage and Matching Metrics.** $\mathcal{C}'$ represents the generated conformation and $\tilde{\mathcal{C}}'$ represents that after Kabsch alignment (Kabsch, 1976), and $n$ denotes the total number of atoms. $\mathcal{C}_i$ and $\tilde{\mathcal{C}}'_i$ correspond to the coordinate vectors of the $i$-th atom in $\mathcal{C}$ and $\tilde{\mathcal{C}}'$, respectively. The symbol $\delta$ is defined as a predetermined RMSD threshold. Additionally, $\mathbf{S}_r$ and $\mathbf{S}_g$ denote the sets of reference and generated conformations, respectively. Based on these definitions, the **Recall** measure is formalized as follows:

$$\text{COV-R}(\mathbf{S}_r, \mathbf{S}_g) = \frac{1}{|\mathbf{S}_r|}\left|\left\{\mathcal{C} \in \mathbf{S}_r | \mathrm{RMSD}(\mathcal{C}, \tilde{\mathcal{C}}') \leq \delta, \exists\, \mathcal{C}' \in \mathbf{S}_g\right\}\right| \tag{26}$$

$$\text{MAT-R}(\mathbf{S}_r, \mathbf{S}_g) = \frac{1}{|\mathbf{S}_r|}\sum_{\mathcal{C}\in\mathbf{S}_r} \min_{\mathcal{C}'\in\mathbf{S}_g} \mathrm{RMSD}(\mathcal{C}, \tilde{\mathcal{C}}') \tag{27}$$

COV-R measures the percentage of conformations in the reference set that are covered by the generated set. Coverage is defined such that, for each conformation in the reference set, there exists at least one conformation in the generated set with an RMSD value smaller than the threshold $\delta$. Conversely, MAT-R calculates the average of the minimum RMSD values between conformations in the reference set and those in the generated set.

The **Precision**, defined by COV-P and MAT-P, mirrors Recall but with $\mathbf{S}_r$ and $\mathbf{S}_g$ swapped. Recall emphasizes diversity, while Precision focuses on quality. Following previous works, the threshold $\delta$ is set to 0.5 Å for QM9 and 1.25 Å for Drugs. $\mathbf{S}_g$ is set to twice the size of $\mathbf{S}_r$.

### B.3. The Neural Vector Field $\boldsymbol{v}_\theta$

As mentioned in Section 4.2, we parameterize the vector field predictor as a neural network $\boldsymbol{v}_\theta$ to learn the true vector field in the *flow matching* training phase. For any $t \in [0, 1]$, with $\mathbf{z}_0 \sim N(0, I)$ and $\mathbf{z}_1 \sim p(\mathbf{z}|\mathcal{G})$, let $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{z}_1$. The true vector field $\boldsymbol{v}_{true} = \mathbf{z}_1 - \mathbf{z}_0$ is predicted using $\boldsymbol{v}_\theta(\mathbf{z}_t, t, \mathcal{G})$.

Following ETFlow (Hassan et al., 2024), $\boldsymbol{v}_\theta$ is parameterized as an enhanced version of TorchMD-NET (Thölke & De Fabritiis, 2022). Readers are encouraged to refer to the relevant sections of the ETFlow paper for a detailed explanation of the network design. In the following sections, the Embedding Layer and Output Layer will be introduced to demonstrate how $\mathbf{z}_t$, $t$, and $\mathcal{G}$ are integrated into the network.

**Embedding Layer.** For each atom in a total of $n$ atoms, let $x_i$ denote the atomic number and $h_i$ represent the atomic attributes (e.g., chirality). An invariant embedding, $\mathbf{emb}_{x_i}$, is computed through the following process:

$$x_i = \textit{Embedding}(x_i);\ h_i = \textit{MLP}(h_i). \tag{28}$$

Then a neighborhood embedding $n_i$ is computed to capture local atomic environment:

$$nei_i = \sum_{j \in \delta(i)} \textit{Embedding}(x_j) \cdot g(d_{ij}, l_{ij}). \tag{29}$$

Here, $\delta(i)$ denotes the set of all neighbors of the $i$-th atom, $d_{ij}$ represents the distance between atoms $i$ and $j$ (derived from the pre-calculated meta conformation $\mathcal{C}_{meta}$), and $l_{ij}$ encodes the edge features (either from a radius-based graph or molecular bonds). The interaction function $g(d_{ij}, l_{ij})$ combines distance and edge information, as described in (Hassan et al., 2024).

Finally, all features are combined to derive the invariant embedding $\textit{Emb}_{x_i}$ through a linear projection:

$$\mathbf{emb}_{x_i} = \textit{Linear}([x_i, h_i, nei_i, t, \mathbf{z}_t]) \tag{30}$$

Where $t$ represents the time step, and $[\cdot, \cdot]$ denotes concatenation. Thus, we successfully introduce $\mathbf{z}_t$ to an invariant input, enabling the network to predict the true vector field at time step $t$.

**Output Layer.** The output layer consists of Gated Equivariant Blocks (Schütt et al., 2018). For each atom, it outputs a scalar embedding $x_i$ and a vector feature $\vec{v_i}$. The scalar embedding $x_i$ is pooled to form a global graph embedding, which is then used as the predicted vector field $\boldsymbol{v}_\theta$:

$$\boldsymbol{v}_\theta = \textit{Mean-Pooling}(x_0, x_1, ..., x_{n-1}) \tag{31}$$

### B.4. Results on the Small-scale Drugs dataset

**Coverage and Matching Metrics.** Tab. 4 presents the Coverage and Precision metrics for the Small-scale Drugs dataset. In particular, CoFM achieves the highest precision in all baselines with 5-step sampling, demonstrating a 3.93% improvement in the mean COV-P value and a 10.14% improvement in the mean MAT-P value. For diversity sampling, comparative results on Recall metrics can be observed after 100-step sampling, with CoFM achieving the second-best MAT-R mean and median values across all baselines.

### B.5. Visualization of Molecular Conformation Generation Results

For clearer visualization, the generated conformations of Small-scale QM9 and Drugs are shown in Figure 4 and Figure 5, respectively. **Input Graph** refers to the topology graph of the input molecule, while **Meta Conf.** represents the pre-generated

*Table 4.* Conformation Generation Results on the Small-scale Drugs dataset.

| Methods | Steps | COV-R (%) ↑ | | MAT-R (Å) ↓ | | COV-P (%) ↑ | | MAT-P (Å) ↓ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| RDKit | 1 | 60.91 | 65.70 | 1.2026 | 1.1252 | 72.22 | 88.72 | 1.0976 | 0.9539 |
| GraphDG | 1 | 08.27 | 00.00 | 1.9722 | 1.9845 | 02.08 | 00.00 | 2.4340 | 2.4100 |
| ConfVAE | 1 | 55.20 | 59.43 | 1.2380 | 1.1417 | 22.96 | 14.05 | 1.8287 | 1.8159 |
| Geomol | 1 | 67.16 | 71.71 | 1.0875 | 1.0586 | - | - | - | - |
| DMCG | 1 | **96.52** | **100.00** | **0.7220** | **0.7161** | **81.05** | **95.51** | **0.9210** | **0.8785** |
| ***3D Space Diffusion*** | | | | | | | | | |
| CGCF | 1000 | 53.96 | 57.06 | 1.2487 | 1.2247 | 21.68 | 13.72 | 1.8571 | 1.8066 |
| ConfGF | 5000 | 62.15 | 70.93 | 1.1629 | 1.1596 | 23.42 | 15.52 | 1.7219 | 1.6863 |
| SDEGen[a] | 1500 | 56.01 | 56.07 | 1.2371 | 1.2303 | 25.07 | 16.11 | 1.7196 | 1.6794 |
| SDEGen[b] | 2000 | 56.59 | 63.58 | 1.2365 | 1.2246 | 26.44 | 17.31 | 1.7046 | 1.6997 |
| SDEGen[c] | 1500 | 26.29 | 10.15 | 1.6011 | 1.6030 | 11.73 | 03.02 | 2.0078 | 1.9853 |
| SDEGen[d] | 6000 | 67.27 | 74.20 | 1.1256 | 1.1289 | 32.25 | 25.65 | 1.6793 | 1.6587 |
| GeoDiff[e] | 1000 | 82.96 | 96.29 | 0.9525 | 0.9334 | 48.27 | 46.03 | 1.3205 | 1.2724 |
| GeoDiff[f] | 5000 | 88.36 | 96.09 | 0.8704 | 0.8628 | 60.14 | 61.25 | 1.1864 | 1.1391 |
| GeoDiff[g] | 5000 | **89.13** | **97.88** | **0.8629** | **0.8529** | 61.47 | 64.55 | 1.1712 | 1.1232 |
| EcConf | 5 | 84.54 | 91.18 | 0.9341 | 0.9264 | 71.40 | 83.17 | 1.0971 | 1.0270 |
| EcConf | 15 | 85.94 | 92.55 | 0.9046 | 0.8905 | **71.63** | **83.44** | **1.0841** | **1.0176** |
| EcConf | 25 | 86.66 | 91.90 | 0.9016 | 0.8869 | 71.36 | 80.62 | 1.0931 | 1.0307 |
| ***SE(3)-Inv Latent Flow Matching (Ours)*** | | | | | | | | | |
| **CoFM** | 5 | 81.30 | 88.71 | 0.8646 | 0.8479 | **84.98** | **97.65** | **0.8276** | **0.7453** |
| **CoFM** | 25 | 85.12 | 91.55 | 0.8293 | 0.7897 | 83.70 | 96.06 | 0.8626 | 0.7760 |
| **CoFM** | 50 | 85.28 | **92.79** | 0.8266 | **0.7866** | 83.45 | 95.26 | 0.8693 | 0.7823 |
| **CoFM** | 100 | **85.92** | 92.53 | **0.8252** | 0.7881 | 83.27 | 94.96 | 0.8725 | 0.7881 |

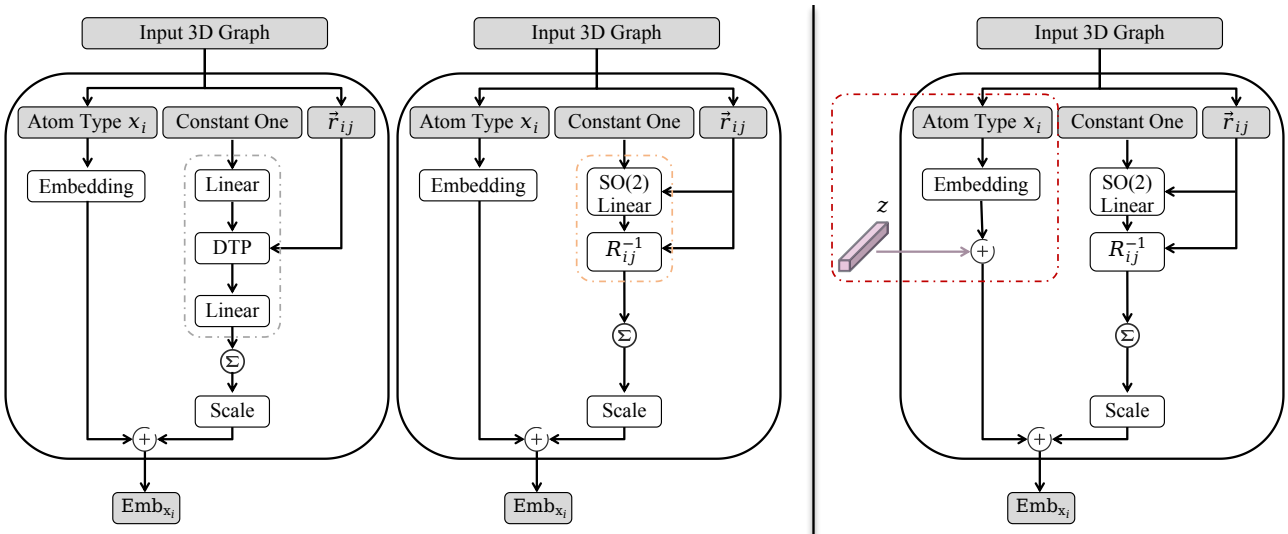
conformation (meta conformation) obtained using RDKit. Notably, for each molecule, only one conformation is selected during preprocessing. **Ref.** denotes the reference conformations, and **Gen.** indicates the conformations generated by CoFM.

### B.6. Model Configuration

**AutoEncoder.** We define the total number of Equiformer blocks in the encoder as $B_{\text{enc}}$. As outlined in Section 4.1, the stacked decoder consists of $p$ small EquiformerV2 networks, with the number of Equiformer blocks in each network denoted as $B_{\text{dec}}$. $l_{\text{max}}$ represents the maximum degree of the irreps features, and $c$ denotes the number of channels for each *type-l* feature. $d$ refers to the dimension of the latent vector. Other hyperparameters remain largely unchanged compared to the original EquiformerV2 (Liao et al., 2023). For further details, please refer to the code implementation (which will be open-sourced upon acceptance).

For both datasets, we set $B_{\text{enc}} = 8$, $p = 2$, and $B_{\text{dec}} = 5$ (as shown in Figure 1), with $l_{\text{max}} = 4$, $c = 256$, and $d = 48$. In the loss function, we set $\lambda_1 = 3$, $\lambda_2 = \lambda_3 = 0.2$, and $\lambda_4 = 1$, with $\alpha = 1$ and $\beta = 0.001$ for both datasets. We use the AdamW (Loshchilov & Hutter, 2018) optimizer with a learning rate of $\eta = 5 \times 10^{-5}$ and other default parameters implemented in the PyTorch toolkit. The best model checkpoint is obtained after training for 100 epochs using a cosine learning rate schedule.

**Vector Field $\boldsymbol{v}_\theta$.** The neural vector field $\boldsymbol{v}_\theta$ is parameterized using a TorchMD-NET (Thölke & De Fabritiis, 2022) model, modified by ETFlow (Hassan et al., 2024), as illustrated in Figure 1 (upper-right) and further detailed in Section B.3. Model hyperparameters are kept consistent with those used in ETFlow. The model is trained for only 100 epochs on each dataset using the Adam optimizer (Kingma & Ba, 2015), with a learning rate of $\eta = 5 \times 10^{-4}$. Notably, during generation, the standard deviation $\sigma_0$ of the Gaussian prior is set to 2.0, 1.5, and 0.8 for the Small-scale QM9, Drugs, and Large-scale QM9 datasets, respectively.




(a). feature embedding of Equiformer (b). feature embedding of EquiformerV2 (c). modified feature embedding of EquiformerV2

*Figure 3.* **The input feature embedding process of the Equiformer Network. (a).** The original input feature embedding in Equiformer (Liao & Smidt, 2022); **(b).** The original input feature embedding in EquiformerV2 (Liao et al., 2023); **(c).** The modified input feature embedding incorporating our SE(3)-invariant latent vector $\mathbf{z}$ (**Ours**).

## C. Important Details of EuiformerV2

In recent research, equivariant networks (Thomas et al., 2018; Batzner et al., 2022; Brandstetter et al., 2021; Liao & Smidt, 2022; Liao et al., 2023) have become dominant tools for modeling 3D atomistic systems, including tasks such as 3D molecular property prediction, pretraining, and force estimation. Among them, **Equiformer**, encompassing both **Equiformer** (Liao & Smidt, 2022) and **EquiformerV2** (Liao et al., 2023), represents a major advancement in the field. It leverages the strengths of Transformer architectures while integrating SE(3)/E(3)-equivariant features through irreducible representations (irreps), enabling more effective and symmetry-aware learning for molecular modeling.

**Embedding.** The input feature embedding process of Equiformer networks integrates atom embedding and edge-degree embedding. The atom embedding, originally implemented as a linear layer to transform the one-hot encoding of atom species, is redefined as a lookup table *Embedding* for the atom type $x_i$. For edge-degree embedding, as illustrated in the right branch of Figure 3(a), a constant one-vector is first transformed into messages encoding local geometry through two linear layers and an intermediate DTP (Depth-Wise Tensor Products) (Howard, 2017) layer. The resulting information is then aggregated using a summation operation to effectively encode the degree information. In EquiformerV2 (Liao et al., 2023), as shown in the right branch of Figure 3(b), the original linear layers and DTP layers are replaced with a single SO(2) linear (Passaro & Zitnick, 2023) layer followed by a rotation matrix $R_{ij}^{-1}$, enhancing the network's ability to capture geometric information more efficiently. Notably, the atom embedding process serves as a fundamental mechanism for seamlessly integrating our SE(3)-invariant latent vector $\mathbf{z}$ into the decoder backbone $\phi_{Dec}$, as illustrated in Equation 9 and highlighted in the red block of Figure 3(c). This integration is formally represented as follows:

$$\mathbf{emb}_{x_i} = \textit{Embedding}(x_i) + \mathbf{z} \tag{32}$$

**Output Head.** As detailed in (Liao et al., 2023), scalar quantities such as energy (the latent vector $\mathbf{z} \in \mathbb{R}^d$ in our case) are predicted using a feed-forward network (*FFN*) that transforms irreducible representations (irreps) features at each node into scalar values (a scalar vector in our case), followed by sum aggregation (*Pooling*) over all nodes. For force prediction (reconstructed conformation $C_{recon} \in \mathbb{R}^{n\times 3}$ in ours case), an equivariant graph attention block is employed, where the *type-l* output $\mathbf{O}_1$ is used as the predicted force for each node.

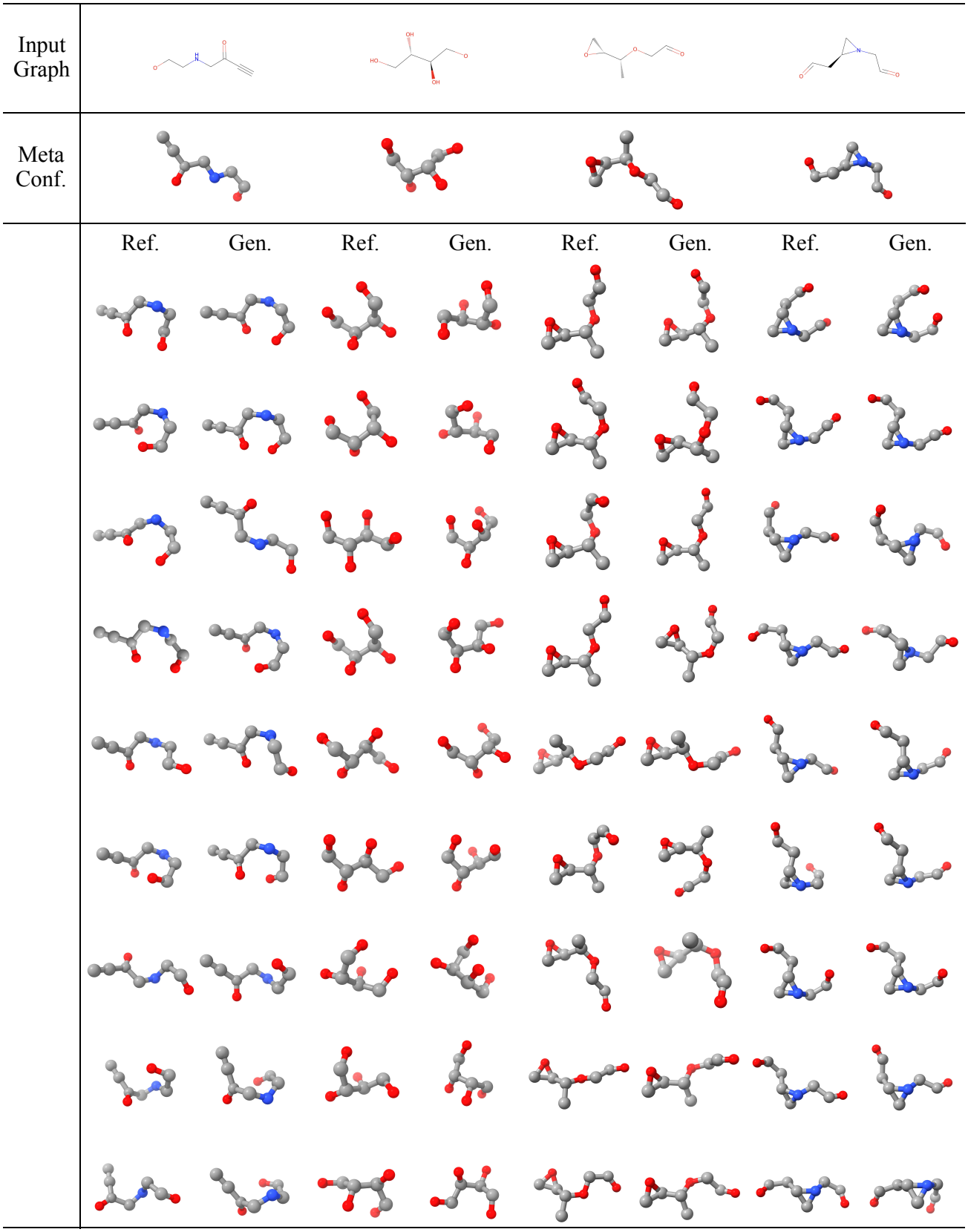



*Figure 4.* Visualization of the molecular conformation generation results on the Small-scale QM9 dataset.

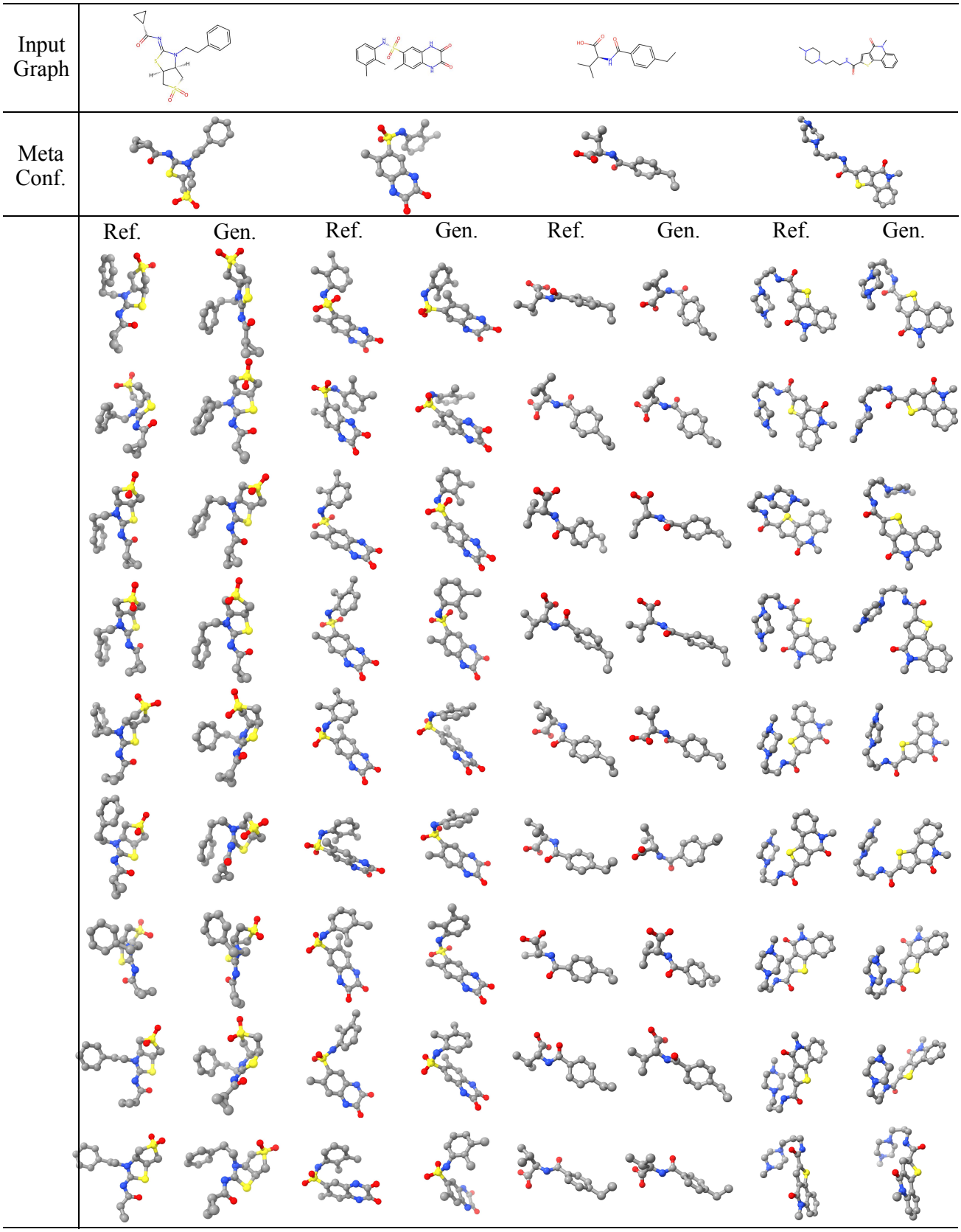



*Figure 5.* Visualization of the molecular conformation generation results on the Small-scale Drugs dataset.