Generative Coarse-Graining of Molecular Conformations

Wujie Wang [1] **Minkai Xu** [2 3] **Chen Cai** [4] **Benjamin Kurt Miller** [5] **Tess Smidt** [1] **Yusu Wang** [4] **Jian Tang** [2 6 7]
Rafael Gómez-Bombarelli [1]

Abstract

Coarse-graining (CG) of molecular simulations simplifies the particle representation by grouping selected atoms into pseudo-beads and drastically accelerates simulation. However, such CG procedure induces information losses, which makes accurate backmapping, i.e., restoring fine-grained (FG) coordinates from CG coordinates, a long-standing challenge. Inspired by the recent progress in generative models and equivariant networks, we propose a novel model that rigorously embeds the vital probabilistic nature and geometric consistency requirements of the backmapping transformation. Our model encodes the FG uncertainties into an invariant latent space and decodes them back to FG geometries via equivariant convolutions. To standardize the evaluation of this domain, we provide three comprehensive benchmarks based on molecular dynamics trajectories. Experiments show that our approach always recovers more realistic structures and outperforms existing data-driven methods with a significant margin.

1. Introduction

Defined as an operation to perform dimension reductions over continuous distributions, coarse-graining (CG) was first proposed by Paul Ehrenfest and Tatyana Afanasyeva in their study of molecular chaos (Ehrenfest & Ehrenfest, 1912). After nearly a century of theoretical developments, CG has become a powerful technique to simplify many complicated problems in physics. Useful applications of CG include the renormalization group techniques for critical phenomena (Kadanoff, 1966; Wilson, 1971) and the usage of collective variables to study reaction dynamics (Laio & Parrinello, 2002). In molecular simulations, CG refers to the simplification of the particle representation by lumping groups of original atoms x into individual beads X with a rule-based CG mapping. CG molecular dynamics (MD) can significantly speed up computational discovery over chemical spaces with simpler combination rules (Marrink et al., 2007; Menichetti et al., 2019); for simulations of large molecules with hundreds of thousands of atoms such as biological and artificial polymers, CG MD allows accessing the long time scales of phenomena like protein folding or polymer reptation (Levitt & Warshel, 1975; Kremer et al., 1988). However, such acceleration comes at the cost of losing fine-grained (FG) atomic details, which are important for studying properties and interactions of atom-level structures or continuing simulations at FG scale. How to accurately recover the FG structure x from CG coordinates X remains a challenging problem.

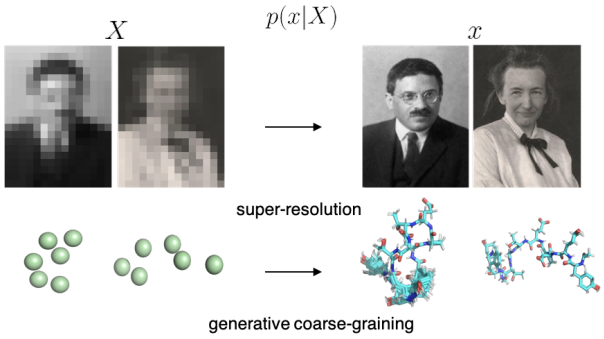

Figure 1: An analogical illustration of the tackled generative coarse-graining problem. **Top**: Image super-resolution recovers higher resolution portraits[1]. **Bottom**: Generative CG generates fine-grained molecular structures.

So far, several methods have been proposed for the backmapping problem based on random projection (Wassenaar et al., 2014) or geometric rules (Shih et al., 2007). However, these methods usually can only produce poor initial geometries which require subsequent restraining molecular dynamics (Shih et al., 2007; Brocos et al., 2012), incurring considerable computational cost and large deviation from the original

[1]Massachusetts Institute of Technology, USA [2]Mila - Québec AI Institute, Canada [3]Universite de Montréal, Canada [4]University of California San Diego, USA [5]University of Amsterdam, Netherlands [6]HEC Montréal, Canada [7]CIFAR AI Chair, Canada. Correspondence to: Rafael Gómez-Bombarelli <rafagb.mit.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]Portraits of P. Ehrenfest and T. Afanasyeva, taken from Dibner Library of the History of Science and Technology.

structures. These methods also require maintaining system-specific fragment libraries that are only applicable to a predefined mapping schema, preventing their usage from broader mapping choices and resolutions. Furthermore, none of these methods are data-driven, and geometries produced by these methods are biased toward predefined fragment geometries. To produce data-informed backampping solutions, several machine learning methods have recently been proposed (Wang & Gómez-Bombarelli, 2019; An & Deshmukh, 2020) to use parameterized functions to deterministically backmap. However, these methods also suffer from low-quality geometries and have not been tested on complex molecular structures. We argue that the problem is very challenging mainly because of the following: **(C1) Stochasticity of backmapping**: Due to surjective nature of CG projections, many different FG configurations can be mapped to the same CG conformation, hence the reverse generative map is one-to-many. However, this vital consideration on stochasticity has been overlooked in current deterministic methods (Brocos et al., 2012; Wassenaar et al., 2014; Wang & Gómez-Bombarelli, 2019; An & Deshmukh, 2020), making it hard to generate diverse structures that capture the underlying FG distributions. **(C2) Geometry consistency**: The backmapped FG coordinates should faithfully represent the underlying CG geometry, which requires that they can be reduced back to the original CG structures. Additionally, given the fact that CG transformation is $E(3)$ equivariant (see details in Section 3.1), the backmapping function should also be $E(3)$ equivariant. A schematic illustration is given in Fig. 2. However, these geometric consistency constraints have not been considered by existing methods. Failure to satisfy these requirements leads to unrealistic FG geometries that are not self-consistent. More discussions can be found in Section 3.2. **(C3) Generality w.r.t mapping protocols and resolutions:** The choice of dimensionality reduction degree for FG to CG mappings varies wildly across CG modeling practices, depending on desired accuracy and simulation speed (Marrink et al., 2007; Bereau & Kremer, 2015; Souza et al., 2021). Therefore, it is desirable to have a general backmapping approach that is compatible with arbitrary CG mapping choices, for better usability under different CG simulation protocols.

To address the requirements and challenges mentioned above, we propose a novel probabilistic model named Coarse-Graining Variational Auto-Encoder (CGVAE), a principled data-driven framework to infer atomistic coordinates from coarse-grained structures. We formulate the stochastic backmapping as a conditional generative task, i.e., modeling the conditional distribution $p_\theta(x|X)$ of FG molecular structures x conditioned on CG structures X. To model the highly complex $p_\theta(x|X)$, we factorize the conditional distribution $p_\theta(x|X)$ as a latent variable model and parameterize a generative map for molecular geometries. This is achieved with $E(3)$ invariant encoding and equivariant decoding operations, using the message passing framework (Gilmer et al., 2017). Our model generates coordinates in a one-shot fashion, requiring no predefined fragment geometry libraries to guide the backmapping.

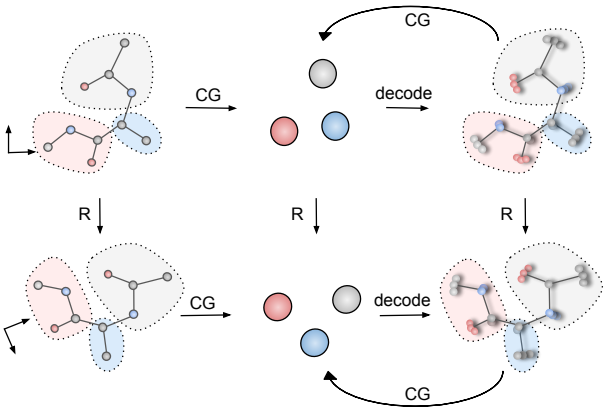

Figure 2: A diagram illustrating the geometric constraints for the backmapping function: 1) The backmapped coordinates should map back to the original CG coordinates; 2) because the CG transformation is equivariant, the backmapping transformation should also be equivariant.

Our contributions can be summarized as follows:

- We provide a principled probabilistic formulation for the backmapping problem of molecular conformations, and propose how to approximate this conditional distribution with a latent variable model in CGVAE (**C2**).
- We design a rigorous and expressive backmapping function that incorporates equivariant convolutions to satisfy the geometry consistency criteria (**C2**).
- CGVAE is CG mapping-agnostic (**C3**) and can be applied to diverse mapping protocols used in MD simulations in practice.
- To facilitate further developments in the field, we propose two benchmark datasets and formulate a suite of metrics to evaluate the quality of generated FG structures.

Experiments on proposed benchmarks show that our model can consistently achieve superior performance compared with previous data-driven methods.

2. Related Works

Reverse algorithms from CG to FG usually require two steps: backmapping projection followed by force field optimization (Guttenberg et al., 2013). The projection typically relies on a deterministic linear backmapping operation (Wassenaar et al., 2014; Rzepiela et al., 2010; Brocos et al., 2012) or a parameterized deterministic function learned by

supervised learning (Wang & Gómez-Bombarelli, 2019; An & Deshmukh, 2020). However, these methods often produce invalid structures because they lack equivariance and chemical rule supervision, and thus rely on intensive force field equilibration of the backmapped systems (Rzepiela et al., 2010; Wassenaar et al., 2014). Recently, probabilistic backmapping (**C1**) has been incorporated to generate FG coordinates, but its use is limited to condensed phase systems with voxelized representation as model inputs (Stieffenhofer et al., 2020). To date, there have not been works that considered **C1-C3** all at once.

3. Preliminaries

3.1. CG Representations of Molecular Structures

We represent fine-grained (FG) systems as atomistic coordinates $x = \{x_i\}_{i=1}^n \in \mathbb{R}^{n\times 3}$. Similarly, the coarse-grained (CG) systems are represented by $X = \{X_i\}_{i=1}^N \in \mathbb{R}^{N\times 3}$ where $N < n$. Let $[n]$ and $[N]$ denote the set $\{1, 2, ..., n\}$ and $\{1, 2, ..., N\}$ respectively. CG operation of molecular simulations can be defined as an assignment $m : [n] \to [N]$, which maps each FG atom i in $[n]$ to CG atom $I \in [N]$, i.e., the CG node I is composed from an ordered set of atoms $C_I = (k \in [n] \mid m(k) = I)$. To ensure that particle mass and momentum are defined consistently with CG (Noid et al., 2008), each atom i can only contribute to at most one I. This CG projection operation can be represented as $X = Mx$. $M \in \mathbb{R}^{N\times n}$ with $M_{I,i} = \frac{w_i}{\sum_{j\in C_I} w_j}$ if $i \in C_I$ and 0 otherwise. Here, w_i is the projection weight of atom i. It can be either the atomic mass i or simply 1, which corresponds to placing X_I at the center of mass or the center of geometry of C_I. The normalization term in M ensures that coarse coordinates are the weighted averages of FG geometries.

3.2. Geometric Requirements of Backmapping

In this part, we discuss the symmetry properties of CG transformation to identify the geometric requirements to design the backmapping decoder function. First, we show that CG projection is $E(3)$ equivariant:

Property 3.1. *Let $f_M : \mathbb{R}^{n\times 3} \to \mathbb{R}^{N\times 3}$ be the linear transformation $f_M(x) := Mx$. f_M is $E(3)$ equivariant, i.e., $f_M(Qx + g) = Qf_M(x) + g$, where Q is a 3×3 orthogonal matrix, and g is a translational vector.*[2]

The proof is provided in Appendix C.3. It states that the CG coordinates $X = Mx$ will always translate, rotate, and reflect in the same way as x, obeying $E(3)$ equivariance. For a generic backmapping function Dec : $\mathbb{R}^{N\times 3} \to \mathbb{R}^{n\times 3}$, it is natural to require that the backmapped coordinate $\widetilde{x}$ can also be mapped back to X for self-consistency, i.e. $M\text{Dec}(X) = X$. As a corollary, such consistency requirements implies that if f_M is equivariant, Dec should also be equivariant. (see Appendix C.4). Formally, we consider the following requirements in our decoder design to ensure the geometric consistency discussed above:

R1. $M\widetilde{x} = M\text{Dec}(X) = X$.
R2. $\text{Dec}(QX + g) = Q\text{Dec}(X) + g$.

To the best of our knowledge, these geometric requirements have not been considered in previous works. In Section 4.3, we introduce how our approach satisfies these requirements.

[2]Following Satorras et al. (2021), here Qx and $x + g$ are shorthands for $\{Qx_1, ..., Qx_n\}$ and $\{g + x_1, ..., g + x_n\}$. The same convention also applies to QX and $X + g$ at the coarse level.

4. Method

In this section, we introduce our probabilistic framework of generating fine-grained (FG) coordinates from coarse-grained (CG) coordinates. We first present an overview of the conditional generation formalism in Section 4.1 and describe the detailed model architecture in Sections 4.2 and 4.3. Then we show the training and sampling protocol in Section 4.4. A notation table is available in Table 1 in the appendix.

4.1. Generative Coarse-Graining Framework

The CG procedure defines a surjective mapping from the FG coordinates x to the CG coordinates X, and we study the inverse problem of recovering x from X. This one-to-many problem is a generation task since the same CG system can backmap to different FG conformations. We propose to learn a parameterized conditional generative model $p_\theta(x|X)$ to approximate the recovering function. We further incorporate the uncertainty of FG coordinates into a continuous latent space z and factorize the conditional distribution as an integral over z, i.e., $p(x|X) = \int p_\theta(x|X, z)p_\psi(z|X)dz$. This integral is known to be intractable (Kingma & Welling, 2013), and we instead maximize its variational lower bound with an approximate amortized posterior distribution $q_\phi(z|X, x)$ (Sohn et al., 2015):

$$\log p(x|X) \geq \underbrace{\mathbb{E}_{q_\phi(z|x,X)} \log p_\theta(x|X, z)}_{\mathcal{L}_{\text{recon.}}} + \underbrace{\mathbb{E}_{q_\phi(z|x,X)} \log \frac{p_\psi(z|X)}{q_\phi(z|x, X)}}_{\mathcal{L}_{\text{reg.}}} \quad (1)$$

where $p_\theta(x|X, z)$ is the decoder model to recover x from X and z, and $q_\phi(z|x, X)$ and $p_\psi(z|X)$ are the encoder and prior model respectively to model the uncertainties from CG reduction. The first term $\mathcal{L}_{\text{recon.}}$ is the expected reconstruction error of generated FG structures, and the second term $\mathcal{L}_{\text{reg.}}$ can be viewed as a regularization over the latent space. A schematic of the whole framework is provided in Fig. 3. In the following sections, we elaborate on how we

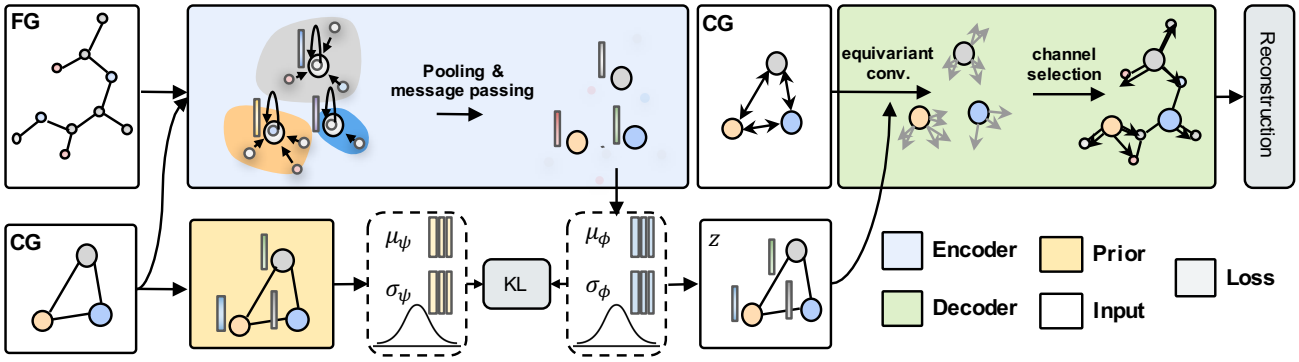

Figure 3: The overall framework of generative CG modeling. The encoder $q_\phi(z|x, X)$ takes both FG and CG structures as inputs, and outputs invariant latent distributions for each CG node via message passing and pooling. Given sampled latent variables and the CG structure, the decoder $p_\theta(x|X, z)$ learns to recover the FG structure through equivariant convolutions. The whole model can be learned end-to-end by optimizing the KL divergence of latent distributions and reconstruction error of generated FG structures.

estimate these distributions with graph neural networks.

4.2. CG Encoding and Prior

Overview of the data representation. The encoding process extracts information from both the FG structure $\mathcal{G}_{\text{FG}}$ and CG structure $\mathcal{G}_{\text{CG}}$. $\mathcal{G}_{\text{FG}}$ represents radius graphs with a distance cutoff d_{cut}. Radius graph is a widely adopted choice that can differentiate conformations with the same molecular graph (Axelrod & Gomez-Bombarelli, 2020).$\mathcal{G}_{\text{CG}}$ are induced graphs based on assignment map m as introduced in Section 3.1. See Appendix D for detailed descriptions of these graphs in our model. In short, the nodes are labeled with atomic types and the edges are labeled with distances.

Estimating the posterior distribution: $(X, x) \to z$. The encoder model extracts CG-level invariant embeddings by three operations at each convolution step: 1) message passing at FG level, 2) pooling operation that maps FG space to CG space, and 3) message passing at CG level. After convolution updates, the obtained CG embeddings are then used to parameterize the latent distribution with CG latent variable $z \in \mathbb{R}^{N\times F}$ where F is the dimension of features. For each operation, we use a SchNet-like architecture (Schütt et al., 2017). The first convolution process at FG level can be described using message passing neural networks (MPNN) (Gilmer et al., 2017):

$$h_i^{t+1} = \text{Update}\Big(\sum_{j\in N(i)} \text{Msg}(h_i^t, h_j^t, \text{RBF}(d_{ij})),\ h_i^t\Big), \quad (2)$$

where $N(i)$ denotes the neighbors of i in $\mathcal{G}_{\text{FG}}$ and d_{ij} denotes the distance between atoms i and j. $\text{RBF}(\cdot)$ is the radial basis transformation introduced in Klicpera et al. (2020); Schütt et al. (2021), which transforms the distances to high dimensional features. The initial node embeddings h_i^0 are parameterized from atomic types. Since both atomic types and edge distances are invariant features, the produced FG-wise features will also be invariant. The FG embeddings are then pooled to CG level based on the CG map m:

$$\widetilde{H}_I^t = \text{Update}\Big(\sum_{i\in C_I} \text{Msg}\left(h_i^{t+1}, H_I^t, \text{RBF}(d_{iI})\right),\ H_I^t\Big), \quad (3)$$

where $\widetilde{H}_I^t \in \mathbb{R}^F$ is the pooled CG embeddings from the FG graph and d_{iI} is the distance between x_i and X_I, i.e., $d_{iI} = \|x_i - X_I\|_2$; H_I^0 is initialized as 0_F. This step pools messages from FG nodes to their assigned CG nodes, followed by an updating procedure. The CG-level node features are further parameterized with message passing at the CG level:

$$H_I^{t+1} = \text{Update}\Big(\sum_{J\in N(I)} \text{Msg}(\widetilde{H}_I^t, \widetilde{H}_J^t, \text{RBF}(d_{IJ})),\ H_I^t\Big), \quad (4)$$

where $N(I)$ denotes the neighbors of I in $\mathcal{G}_{\text{CG}}$, and d_{IJ} denotes the distance between CG beads I and J.

The aforementioned three steps constitute the convolutional module for extracting both CG and FG level information, with their detailed design described in Appendix E.1. After several steps of convolutional updates in our encoder, we can obtain the final embeddings $\mathbf{Enc}_\phi(x, X) = H_I^\phi \in \mathbb{R}^F$ for each CG atom I. Then two feedforward networks μ_ϕ and σ_ϕ are applied to parameterize the posterior distributions as diagonal multivariate Gaussians $\mathcal{N}(z_I|\mu_\phi(H_I^\phi), \sigma_\phi(H_I^\phi))$. Note that, H^ϕ are designed as invariant features and thus the resulting likelihood is also invariant.

Estimating the prior distribution: $X \to z$. The prior model $\mathbf{Prior}_\psi(X)$ only takes the CG structure $\mathcal{G}_{\text{CG}}$ as inputs to model the latent space. The initial CG node features are set as the pooled sum of one-hot FG fingerprints $H_I^0 = \sum_{i\in C_I} h_i^0$. Then we employ the same networks as

defined in Eq. (4) to conduct graph convolutions over $\mathcal{G}_{CG}$ and update the node embeddings. After obtaining the final CG embeddings $\mathbf{Prior}_\psi(X) = H^\psi$, we also model $p_\psi(z|X)$ as the products of CG-wise normal distributions $\mathcal{N}(z_I|\mu_\psi(H_I^\psi), \sigma_\psi(H_I^\psi)$ with means and diagonal covariances parameterized from H_I^ψ using two separate neural networks μ_ψ and σ_ψ.

4.3. Multi-channel Equivariant Decoding

Our decoder design is inspired by recent advances in vector-based graph neural nets (Schütt et al., 2021; Satorras et al., 2021; Jing et al., 2020; Batzner et al., 2021). On the high level, the decoder $p_\theta(\widetilde{x}|X, z)$ maps the coarse geometry $X \in \mathbb{R}^{N\times 3}$ and invariant latent variables $z \in \mathbb{R}^{N\times F}$ to produce $\widetilde{x} \in \mathbb{R}^{n\times 3}$ as predictions of FG geometry x. Our proposed decoder consists of three steps 1) Given X and invariant feature z from the encoder, we will generate F equivariant vector channel for each bead, resulting in $V^\theta \in \mathbb{R}^{N\times F\times 3}$. 2) As each bead corresponds to different number of atoms, not all channels will be used. From vector channels V^θ, we select a proper number of channels for each bead to form $\Delta\widetilde{x} \in \mathbb{R}^{n\times 3}$ which is the prediction for the relative position Δx. 3) In the last step, we go from relative position $\Delta\widetilde{x}$ to absolute $\widetilde{x}$. This step has to satisfy **R1**. We elaborate on these three steps next.

Equivariant convolution: $(X, z) \to V^\theta$**.** We use equivariant convolutions based on inter-node vectors to predict $\Delta\widetilde{x}$. Apart from invariant node and edge features, our decoder takes a set of edge-wise equivariant vector features $\{\widehat{E}_{IJ} = \frac{X_J - X_I}{d_{IJ}} \mid (I, J) \in \mathcal{E}_{\text{CG}}\}$ as inputs, which form a basis set to construct $\Delta\widetilde{x}$. Additionally, our model also include pseudoscalars/pseudovectors which contain additional information of chirality. Unlike scalars/vectors, they undergo a sign flip under reflection transformation (see Appendix E.2.1). The message passing operations in the decoder update node-wise scalar features $H \in \mathbb{R}^F$, pseudoscalar features $\overline{H} \in \mathbb{R}^F$, vector features $V \in \mathbb{R}^{F\times 3}$, and pseudovector features $\overline{V} \in \mathbb{R}^{F\times 3}$ separately. The convolutional updates in our decoder are performed as follows:

$$
\begin{aligned}
\Delta H_I^t &= \sum_{J\in N(i)} W_1 \circ H_J^t \\
\Delta \overline{H}_I^t &= \sum_{J\in N(i)} V_I^t \cdot \overline{V}_J^t \\
\Delta V_I^t &= \sum_{J\in N(i)} \Big(W_2 \circ (V_I^t \times \overline{V}_J^t) + W_3 \circ \overline{H}_I^t \circ \overline{V}_J^t \\
&\qquad + W_4 \widehat{E}_{IJ} + W_5 \circ V_J^t\Big) \\
\Delta \overline{V}_I^t &= \sum_{J\in N(i)} \Big(W_6 \circ (V_I^t \times V_J^t) + W_7 \circ (\overline{V}_I^t \times \overline{V}_J^t) \\
&\qquad + W_8 \circ \overline{V}_J^t + W_9 \circ \overline{H}^t \circ V_J^t\Big)
\end{aligned}
\tag{5}
$$

where $\times$, $\circ$ and $\cdot$ are cross product, element-wise product and dot product operations respectively. W_1-W_9 are parameterized invariant edge-wise filters. Let $L_1 : \mathbb{R}^F \to \mathbb{R}^F$ and $L_2 : \mathbb{R}^K \to \mathbb{R}^F$ be neural networks, then each filter is implemented as as $W \in \mathbb{R}^F = L_1(\text{RBF}(d_{IJ})) \circ L_2(H_J)$. The message passing operations are followed by an equivariant update block $\sigma(\cdot, \cdot)$ as proposed in PaiNN(Schütt et al., 2021). It is used as an update block that linearly mixes vector channels and introduces non-linear coupling between H and V. For pseudoscalar and pseudovector $\overline{H}$ and $\overline{V}$, we simply apply a residual update to ensure that they flip sign under reflection:

$$
\begin{aligned}
H_I^{t+1}, V_I^{t+1} &= \sigma(H_I^t + \Delta H_I^t, V_I^t + \Delta V_I^t) \\
\overline{H}_I^{t+1}, \overline{V}_I^{t+1} &= \overline{H}_I^t + \Delta \overline{H}_I^t, \overline{V}_I^t + \Delta \overline{V}_I^t
\end{aligned}
\tag{6}
$$

Pseudoscalars and pseudovectors in our decoder come from cross product updates. This introduces a richer basis set for the coordinate construction especially for low N cases. When $N = 3$, the span of the vector basis will be constrained in a plane, and cross product can overcome this limitation by introducing a vector basis in the orthogonal directions, increasing the expressiveness of the model.

The initial H_I^0 is obtained from $z \sim q_\phi(z|X, x)$ for training, or $p_\psi(z|X)$ for sampling. We initialize other feature set with zero vectors, i.e., $\overline{H}_I^0 = 0_F$, $V_I^0 = 0_{F\times 3}$, and $\overline{V}_I^0 = 0_{F\times 3}$. For 3-bead CG geometries which are not chiral, we initialize $\overline{H}_I^0 = 1_F$ to break the symmetry (see Appendix F.5 for more discussions). In this way, information in the pseudovector channel can be propagated to vector channels to construct a basis set that spans 3D.

After several layers of convolution updates, the decoder produces the final vector output $V^\theta \in \mathbb{R}^{N\times F\times 3}$. It is easy to show that V^θ transforms in an $E(3)$ equivariant way because it is updated strictly with vector operations, which satisfies **R2** in Section 3.2. More discussions on the equivariance properties of the decoder are available at Appendix E.2 and Schütt et al. (2021).

Channel selection: $V^\theta \to \Delta\widetilde{x}$**.** In this part we introduce how to obtain the relative position predictions $\Delta\widetilde{x} \in \mathbb{R}^{n\times 3}$ from the equivariant vector outputs $V^\theta \in \mathbb{R}^{N\times F\times 3}$. This requires a channel selection procedure illustrated in Fig. 4A. As each atom i is assigned to $I = m(i)$, the corresponding prediction of position $\Delta\widetilde{x}_i$ is chosen to be the vector channel with the index of i in C_I. This requires an index retrieval function $\text{Index}(i, C_I)$ given that C_I is ordered.[3] The prediction of $\Delta\widetilde{x}$ does not utilize all the $F > |C_I|$ vector channels at the final vector output V^θ, while all the channels participate in the convolutions before the read-out. We select a proper number of vector channels from V^θ and concatenate them to form the FG level prediction $\Delta\widetilde{x}$. More

[3] $\text{Index}(i, C_I)$ retrieves the index of $i \in C_I$. For example, $\text{Index}(1, (1, 2, 4)) = 0, \text{Index}(4, (1, 2, 4)) = 2$

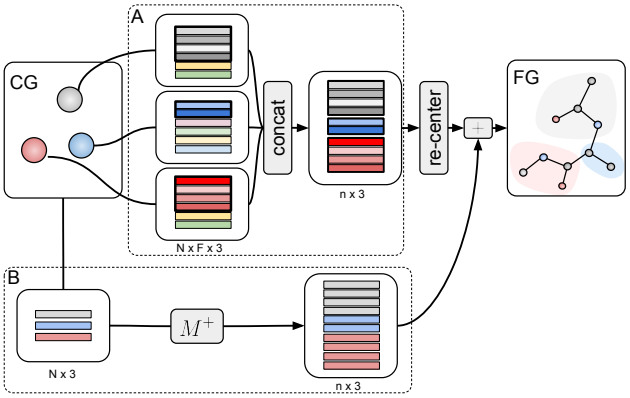

Figure 4: A schematic diagram to describe how coordinates are reconstructed from vector channels. **A.** The prediction of $\Delta\widetilde{x}_i$ are compiled from selected vector channels of CG node $m(i)$ based on their index in set C_I; this is followed by the concatenation of selected vectors to compile $\Delta\widetilde{x} = \bigoplus_i V^{\theta}_{m(i),\text{Index}(i,C_I)}$. The obtained coordinates are further re-centered by subtracting $M^+M\Delta\widetilde{x}$ so that **R2** is satisfied. **B.** The CG coordinates are "lifted" to the FG space via the lifting operator M^+ to get $M^+X \in \mathbb{R}^{n\times 3}$ which are used as the geometric reference for constructions.

precisely, $\Delta\widetilde{x} = \bigoplus_i V^{\theta}_{m(i),\text{Index}(i,C_I)}$ where $m(i)$ selects CG beads and $\text{Index}(i, C_I)$ selects vector channel of bead I. This operation can be done conveniently in Pytorch (Paszke et al., 2019). See implementation details in Appendix E.2.3.

Compile predictions for FG coordinates: $\Delta\widetilde{x} \to \widetilde{x}$. As $\Delta\widetilde{x}$ is the relative position, we need one more step to go from $\Delta\widetilde{x}$ to final FG coordinates $\widetilde{x}$. This is done by setting $\widetilde{x} := M^+X + \Delta\widetilde{x} - M^+M\Delta\widetilde{x}$. $M^+ \in \mathbb{R}^{n\times N}$ is the lift operator with $M^+_{i,I} = 1$ if $I = m(i)$ and 0 otherwise (Appendix C.1). Similar constructions can also be found in graph-coarsening methods (Loukas, 2019; Cai et al., 2020). As described by Fig. 4B, M^+ maps the point sets space from $\mathbb{R}^{N\times 3}$ to $\mathbb{R}^{n\times 3}$ by assigning or "lifting" CG coordinates back to their contributing FG atoms in C_I. The term $-M^+M\Delta\widetilde{x}$ is added to re-center $\Delta\widetilde{x}$ so that we can get the original X back after CG projection in order to satisfy **R1**. See Appendix E.2.4 a formal statement with proof.

4.4. Model Training and Sampling

Reconstruction loss calculation. We calculate the reconstruction loss introduced in Eq. (1) as two invariant components: $\mathcal{L}_{\text{MSD}}$ and $\mathcal{L}_{\text{graph}}$. The mean square distance loss $\mathcal{L}_{\text{MSD}} = \frac{1}{n}\sum_{i=1}^{n} \|\widetilde{x}_i - x_i\|_2^2$ is a standard measurement to quantify differences between the generated reference coordinates (Wang & Gómez-Bombarelli, 2019; Xu et al., 2021). To supervise the model to produce valid distance geometries, we introduce another loss term $\mathcal{L}_{\text{graph}}$. Let $\mathcal{E}$ denote the set of chemical bonds in the generated molecular graph. Following previous works (Xu* et al., 2021), we also expand the set with some auxiliary multi-hop edges to fix rotatable bonds. We also include all 2-hop neighbor pairs in $\mathcal{E}$. Then the objective can be calculated as $\mathcal{L}_{\text{graph}} = \frac{1}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}}(d(\widetilde{x}_i, \widetilde{x}_j) - d(x_i, x_j))^2$, which can be viewed as a supervision over the bond lengths. The total reconstruction loss is computed as $\mathcal{L}_{\text{recon.}} = \mathcal{L}_{\text{MSD}} + \gamma\mathcal{L}_{\text{graph}}$, where γ is a weighting hyperparameter to balance the two loss contributions.

Training and sampling. We adopt the training objective $\mathcal{L} = \mathcal{L}_{\text{recon.}} + \beta\mathcal{L}_{\text{reg.}}$, where β is a hyperparameter to control the regularization strength (Higgins et al., 2017). During training, the CG latent variable z is sampled from $q_\phi(z|x, X)$ by $z = \mu_\phi + \sigma_\phi \circ \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. Then both z and X are taken as inputs of the decoding model to generate $\widetilde{x}$. For sampling, given the coarse structure X, we first sample the invariant latent variable from the prior model by $z \sim p_\psi(z|X)$, and then pass the sampled latent representation z and the coarse coordinates X into the decoder to generate FG coordinates $\widetilde{x} = \mathbf{Dec}_\theta(X, z)$.

5. Experiments

In this section, we introduce our experimental settings and discuss both the quantitative and qualitative results. We first introduce the benchmark datasets and comprehensive metrics in Section 5.1 and Section 5.2, and then visualize and analyze the results in Section 5.3.

5.1. Benchmarks

Datasets. We evaluate our models on 2 datasets that have been used as benchmarks for data-driven CG modeling ((Wang & Gómez-Bombarelli, 2019; Wang et al., 2019; Husic et al., 2020)): MD trajectories of alanine dipeptide and chignolin. The details of the simulation protocol can be found in Husic et al. (2020). Alanine dipeptide is a classical benchmark system for developing enhanced sampling methods for molecular conformation, containing 22 atoms. Chignolin is a 10-amino acid mini-protein with 175 atoms and features a β-hairpin turn in its folded state. It is a more challenging structure to evaluate our model's capacity of decoding complex conformations. For alanine dipeptide and chignolin, we randomly select 20,000 and 10,000 conformations respectively from the data trajectories and perform 5-fold cross-validation to compute mean and standard errors of evaluation metrics.

Baselines. We compare our model to two baseline models proposed in recent works. The first method is a learnable **linear** backmap matrix (Wang & Gómez-Bombarelli, 2019). The method itself is $O(3)$ equivariant, and we fix the translation freedom by re-centering the geometry to its geometric center for each sample. This method is a data-driven and equivariant extension of the approach described

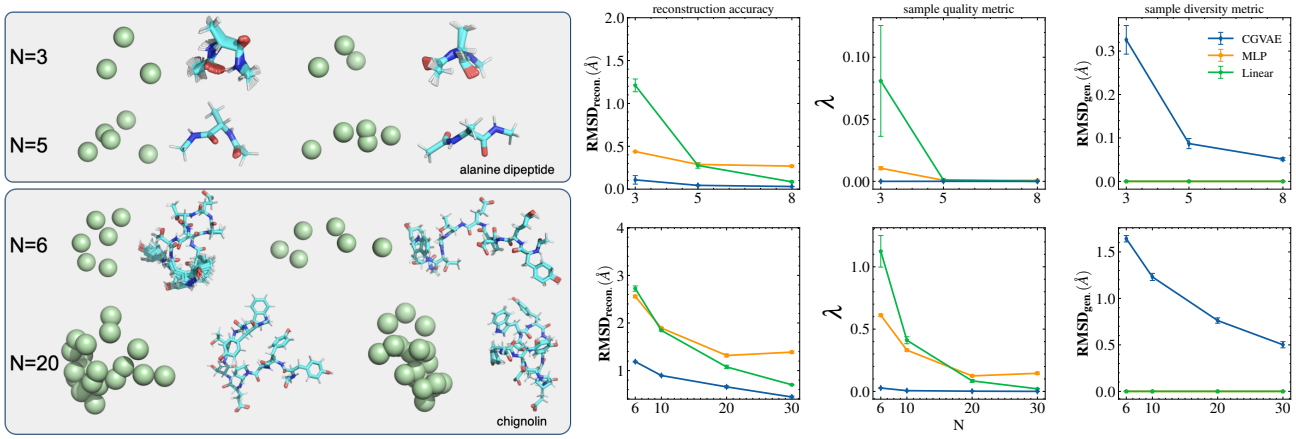

Figure 5: **Left**: Examples of CG and generated FG geometries of alanine dipeptide (top) and chignolin (bottom) generated by CGVAE. The geometry visualizations are created by PyMol (DeLano). **Right**: Benchmarks of reconstructed and generated geometries for heavy-atom structures with different resolution for alanine dipeptide (top) and chignolin (bottom).

in Wassenaar et al. (2014), which uses linear and deterministic (not learnable) projection followed by a random displacement. The second baseline employs Multi-Layer Perceptrons (**MLP**) (An & Deshmukh, 2020) to learn the projection. The model transforms the CG coordinates input and the FG coordinates output as 1D flattened arrays, shaped $3N$ and $3n$ respectively. The method uses neural networks to learn the mapping of the flattened vectors, which is not equivariant. Note that, both baselines lack the probabilistic formulation and are deterministic, which suffer from poor sampling diversity.

CG mapping generations. Our method is compatible with arbitrary mapping protocols. For training our CGVAE and baseline models, in addition to the FG structures provided in the datasets, we also need a CG mapping to produce the CG coordinates. In this paper, we use AutoGrain proposed in (Wang & Gómez-Bombarelli, 2019), which learns the CG assignment via the objective function that penalizes distant FG atoms being assigned to the same CG atoms. In practice, we first train the AutoGrain model on each dataset, and then take the learned assignment to compute the CG coordinates X from FG coordinates x. More details about CG mapping generation can be found in Appendix F.2.

5.2. Metrics

Ideally, the generated FG samples should match the original FG molecular structures (quality) and consist of novel conformations (diversity). We propose evaluation metrics for generation and reconstruction tasks that measure model fitting capacity, sample quality, and sample diversity at different CG resolutions N. More detailed descriptions of the proposed evaluation metrics are in Appendix F.3. We evaluate the reconstructed and sampled 3D geometries for both all-atom and heavy-atom (excluding hydrogen atoms) structures because heavy-atom encodes more important structural dynamics.

Reconstruction accuracy. The reconstruction task evaluates the model's capacity to encode and reconstruct coordinates. Given the reference and reconstructed FG structures, we report their root mean squared distance ($\text{RMSD}_{\text{recon}}$). A lower $\text{RMSD}_{\text{recon}}$ indicates the more powerful learning capacity of the model for reconstructing FG geometries.

Sample quality. We evaluate the sample quality by measuring how well the generated geometries preserve the original chemical bond graph. Specifically, we report the graph edit distance ratio $\lambda(\mathcal{G}_{\text{gen}}, \mathcal{G}_{\text{mol}})$, which measures the relative difference between the connectivity of the generated geometries $\mathcal{G}_{\text{gen}}$ and the ground truth bond graph $\mathcal{G}_{\text{mol}}$. For each generated conformation, $\mathcal{G}_{\text{gen}}$ is deduced from the coordinates by connecting bonds between pair-wise atoms where the distances are within a threshold defined by atomic covalent radius cutoff (see Appendix D). The lower $\lambda(\mathcal{G}_{\text{gen}}, \mathcal{G}_{\text{mol}})$ is, the better the $\mathcal{G}_{\text{gen}}$ resembles $\mathcal{G}_{\text{mol}}$, with $\lambda = 0$ indicating $\mathcal{G}_{\text{mol}}$ and $\mathcal{G}_{\text{gen}}$ are isomorphic.

Sample diversity. This metric measures the model capacity to explore the large conformation space. To evaluate sample diversity, we repeatedly sample multiple $z \sim P(z|X)$ and generate an ensemble of geometries using $\mathbf{Dec}_\theta(z, X)$. We compare chemically valid generated geometries with the reference FG geometries using RMSD, and term this metric RMSD_{gen}. Similar ideas have been adopted in related conformation generation literature (Mansimov et al., 2019). A higher value of RMSD_{gen} indicates the better diversity of the geometries generated by the model. Note that, since both baselines are deterministic without the capacity to model sampling diversity, their RMSD_{gen} are zeros.

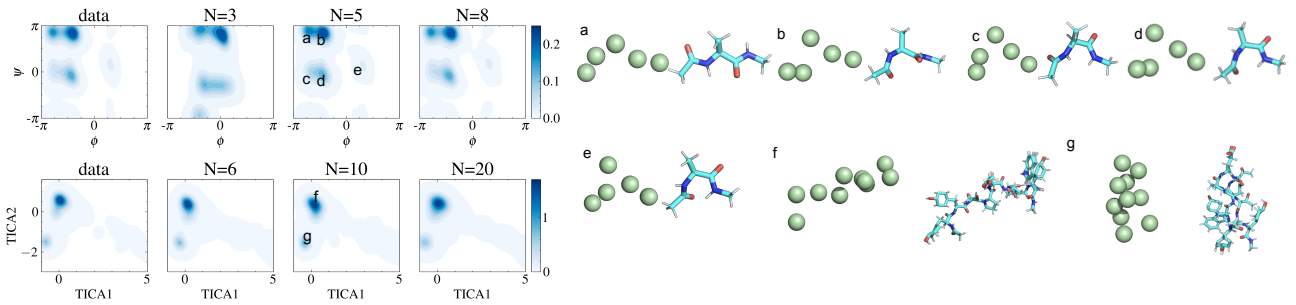

Figure 6: **Left:** 2D Ramachandran plots (top) and TICA plots (bottom) for conformations of alanine dipeptide and chignolin respectively. The structures for the plots are references in the dataset (first column) and samples generated by CGVAE trained with different N (the rest three columns). Figures show that the generated samples recover the important meta-stable states and the distributions agree well with the ground truth. It also shows that the higher resolution model shows better resemblance with the ground truth data. **Right:** Visualization of representative samples in the 2D plots.

5.3. Results and Analysis

Quantitative results. We test our CGVAE models against baseline methods on both datasets. A summary of all the metrics regarding the heavy-atom geometries is reported in Fig. 5. Results for all-atom geometries are left in the Appendix F.4. As shown in Fig. 5, CGVAE models can consistently achieve better reconstruction accuracy, sample quality, and diversity than linear and MLP models among all resolutions. The linear model ensures equivariance but lacks the expressive capacity to encode complex conformations. The MLP model enjoys higher expressiveness, but the lack of equivariance limits its generalization to different roto-translational states. Combining equivariance and MPNNs, CGVAEs can generate FG samples with high quality and diversity. Moreover, we observe that higher-resolution CG representations lead to better reconstruction accuracy, as channels for each CG node are responsible for fewer atoms. Besides, the higher RMSD_{gen} results of the lower resolution models indicate that they can generate more diverse samples because the distribution of z encodes a larger conformational space. Surprisingly we notice that the complex geometries can be well represented with a very small set of CG atoms. Our model works well even for the extremely coarse order reduction of 22-to-3 and 175-to-6 for alanine dipeptide and chignolin respectively. Such extent of CG reduction means potential acceleration of all-atom simulations by one or two orders of magnitude if a $\mathcal{O}(n \log n)$ algorithm such as Particle Mesh Ewald (PME) is used (Darden et al., 1993).

Analyzing generated conformations. To gain a better qualitative understanding of the generated samples, we project the generated samples onto lower-dimensional reaction coordinates for visualization. The visualizations are shown in Fig. 6. For alanine dipeptide samples, we visualize the histogram of the generated samples on $\phi \times \psi$ which are two special central dihedral angles. The resulting histogram is called the Ramachandran plot, which is a classic visualization for analyzing peptide conformations (Ramachandran et al., 1963). The distribution of these two dihedral angles is an important observable because the heavy-atom dynamics of the central alanine are completely described by ϕ and ψ (Feig, 2008). As shown in the plot, with only 3 CG atoms, the model can nearly recover all the important meta-stable states, even though the dihedral angles can only be fully described by at least 6 heavy atoms at the center of the backbone. For analysis of generated chignolin samples, we perform Time-structure Independent Components Analysis (TICA), a popular analysis method to project dynamics onto slow degrees of freedom (Pérez-Hernández et al., 2013; Scherer et al., 2015). It is a useful analysis to project high-dimensional complex conformation variations onto a 2D space to identify meta-stable states of molecular geometries. To conduct the analysis, we use pair distances for backbone atoms (45 atoms, 990 distances) with a lag time of 10 ns for both the ground-truth data and generated samples. We first parameterize the TICA transformation on the simulation data by PyEmma (Scherer et al., 2015), and apply the transformation on the generated geometry sets to produce a plot for comparison. As shown in the figure, our TICA density plot made from generated samples agrees well with the reference data, which demonstrates our model accurately captures the distribution of conformational states.

6. Conclusion

In this work, we present the Coarse-Grained Variational Auto-Encoder (CGVAE), a principled probabilistic model to perform stochastic reverse generation to recover all-atom coordinates from coarse-grained molecular coordinates. Built on equivariant operations, our model encodes all-atom information into invariant latent variables which are sampled to

generate coordinates equivariantly. Comprehensive experiments demonstrate that CGVAE significantly outperforms existing methods. For future work, we plan to extend our general framework to the more complex condensed phase systems which demand accurate backmapping at both intra-molecular and inter-molecular level. We hope that our efforts further improve the predictive power of CG simulations for accelerated chemical and materials discovery.

Code Availability

A reference implementation can be found at https://github.com/wwang2/CoarseGrainingVAE

Acknowledgement

This work is supported by Toyota Research Institute and Google Faculty Award.

References

An, Y. and Deshmukh, S. A. Machine learning approach for accurate backmapping of coarse-grained models to all-atom models. *Chemical Communications*, 56(65): 9312–9315, 2020.

Axelrod, S. and Gomez-Bombarelli, R. Molecular machine learning with conformer ensembles. *arXiv preprint arXiv:2012.08452*, 2020.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *arXiv preprint arXiv:2101.03164*, 2021.

Bereau, T. and Kremer, K. Automated parametrization of the coarse-grained martini force field for small organic molecules. *Journal of chemical theory and computation*, 11(6):2783–2791, 2015.

Bevilacqua, B., Frasca, F., Lim, D., Srinivasan, B., Cai, C., Balamurugan, G., Bronstein, M. M., and Maron, H. Equivariant subgraph aggregation networks. *arXiv preprint arXiv:2110.02910*, 2021.

Brocos, P., Mendoza-Espinosa, P., Castillo, R., Mas-Oliva, J., and Pineiro, Á. Multiscale molecular dynamics simulations of micelles: coarse-grain for self-assembly and atomic resolution for finer details. *Soft Matter*, 8(34): 9005–9014, 2012.

Cai, C., Wang, D., and Wang, Y. Graph coarsening with neural networks. In *International Conference on Learning Representations*, 2020.

Cai, C., Vlassis, N., Magee, L., Ma, R., Xiong, Z., Bahmani, B., Wong, T.-F., Wang, Y., and Sun, W. Equivariant geometric learning for digital rock physics: estimating formation factor and effective permeability tensors from morse graph. *arXiv preprint arXiv:2104.05608*, 2021.

Cohen, T. S., Weiler, M., Kicanaoglu, B., and Welling, M. Gauge equivariant convolutional networks and the icosahedral cnn. *arXiv preprint arXiv:1902.04615*, 2019.

Darden, T., York, D., and Pedersen, L. Particle mesh ewald: An n log (n) method for ewald sums in large systems. *The Journal of chemical physics*, 98(12):10089–10092, 1993.

DeLano, W. Pymol. URL `http://www.pymol.org/pymol`.

Ehrenfest, P. and Ehrenfest, T. *The Conceptual Foundations of the Statistical Approach in Mechanics*. North-Holland Publishing Company, 1912.

Feig, M. Is alanine dipeptide a good model for representing the torsional preferences of protein backbones? *Journal of Chemical Theory and Computation*, 4(9):1555–1564, 2008.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. *arXiv preprint arXiv:2002.12880*, 2020.

Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

Gebauer, N. W., Gastegger, M., and Schütt, K. T. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *arXiv preprint arXiv:1906.00957*, 2019.

Geiger, M., Smidt, T., M., A., Miller, B. K., Boomsma, W., Dice, B., Lapchevskyi, K., Weiler, M., Tyszkiewicz, M., Batzner, S., Uhrin, M., Frellsen, J., Jung, N., Sanborn, S., Rackers, J., and Bailey, M. Euclidean neural networks: e3nn, 2020. URL https://doi.org/10.5281/zenodo.5292912.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Guttenberg, N., Dama, J. F., Saunders, M. G., Voth, G. A., Weare, J., and Dinner, A. R. Minimizing memory as an objective for coarse-graining. *The Journal of chemical physics*, 138(9):094111, 2013.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Husic, B. E., Charron, N. E., Lemm, D., Wang, J., Pérez, A., Majewski, M., Krämer, A., Chen, Y., Olsson, S., de Fabritiis, G., et al. Coarse graining molecular dynamics with graph neural networks. *The Journal of Chemical Physics*, 153(19):194101, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jensen, J. H. xyz2mol. https://github.com/jensengroup/xyz2mol, 2021.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J., and Dror, R. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

Kadanoff, L. P. Scaling laws for ising models near t_c. *Physics Physique Fizika*, 2(6):263, 1966.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kirkwood, J. G. Statistical mechanics of fluid mixtures. *The Journal of chemical physics*, 3(5):300–313, 1935.

Klicpera, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

Kremer, K., Grest, G. S., and Carmesin, I. Crossover from rouse to reptation dynamics: A molecular-dynamics simulation. *Physical review letters*, 61(5):566, 1988.

Laio, A. and Parrinello, M. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20): 12562–12566, 2002.

Landrum, G. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

Levitt, M. and Warshel, A. Computer simulation of protein folding. *Nature*, 253(5494):694–698, 1975.

Loukas, A. Graph reduction with spectral and cut guarantees. *J. Mach. Learn. Res.*, 20(116):1–42, 2019.

Mansimov, E., Mahmood, O., Kang, S., and Cho, K. Molecular geometry prediction using a deep generative graph neural network. *Scientific reports*, 9(1):1–13, 2019.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Maron, H., Litany, O., Chechik, G., and Fetaya, E. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pp. 6734–6744. PMLR, 2020.

Marrink, S. J., Risselada, H. J., Yefimov, S., Tieleman, D. P., and De Vries, A. H. The martini force field: coarse grained model for biomolecular simulations. *The journal of physical chemistry B*, 111(27):7812–7824, 2007.

McGibbon, R. T., Beauchamp, K. A., Harrigan, M. P., Klein, C., Swails, J. M., Hernández, C. X., Schwantes, C. R., Wang, L.-P., Lane, T. J., and Pande, V. S. Mdtraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophysical Journal*, 109(8):1528 – 1532, 2015. doi: 10.1016/j.bpj.2015.08.015.

Menichetti, R., Kanekal, K. H., and Bereau, T. Drug–membrane permeability across chemical space. *ACS central science*, 5(2):290–298, 2019.

Noid, W., Chu, J.-W., Ayton, G. S., Krishna, V., Izvekov, S., Voth, G. A., Das, A., and Andersen, H. C. The multiscale coarse-graining method. i. a rigorous bridge between atomistic and coarse-grained models. *The Journal of chemical physics*, 128(24):244114, 2008.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Pérez-Hernández, G., Paul, F., Giorgino, T., De Fabritiis, G., and Noé, F. Identification of slow molecular order parameters for markov model construction. *The Journal of chemical physics*, 139(1):07B604_1, 2013.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Ramachandran, G., Ramakrishnan, C., and Sasisekharan, V. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95–99, 1963.

Rzepiela, A. J., Schäfer, L. V., Goga, N., Risselada, H. J., De Vries, A. H., and Marrink, S. J. Reconstruction of atomistic details from coarse-grained structures. *Journal of computational chemistry*, 31(6):1333–1343, 2010.

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021.

Scherer, M. K., Trendelkamp-Schroer, B., Paul, F., Pérez-Hernández, G., Hoffmann, M., Plattner, N., Wehmeyer, C., Prinz, J.-H., and Noé, F. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *Journal of Chemical Theory and Computation*, 11:5525–5542, October 2015. ISSN 1549-9618. doi: 10.1021/acs.jctc.5b00743. URL `http://dx.doi.org/10.1021/acs.jctc.5b00743`.

Schütt, K. T., Kindermans, P.-J., Sauceda, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv preprint arXiv:1706.08566*, 2017.

Schütt, K. T., Unke, O. T., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.

Shi, C., Luo, S., Xu, M., and Tang, J. Learning gradient fields for molecular conformation generation. *arXiv preprint arXiv:2105.03902*, 2021.

Shih, A. Y., Freddolino, P. L., Sligar, S. G., and Schulten, K. Disassembly of nanodiscs with cholate. *Nano letters*, 7 (6):1692–1696, 2007.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491, 2015.

Souza, P. C., Alessandri, R., Barnoud, J., Thallmair, S., Faustino, I., Grünewald, F., Patmanidis, I., Abdizadeh, H., Bruininks, B. M., Wassenaar, T. A., et al. Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nature methods*, 18(4):382–388, 2021.

Stieffenhofer, M., Wand, M., and Bereau, T. Adversarial reverse mapping of equilibrated condensed-phase molecular structures. *Machine Learning: Science and Technology*, 1(4):045014, 2020.

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.

Unke, O. T., Bogojeski, M., Gastegger, M., Geiger, M., Smidt, T., and Müller, K.-R. Se (3)-equivariant prediction of molecular wavefunctions and electronic densities. *arXiv preprint arXiv:2106.02347*, 2021.

Wang, J., Olsson, S., Wehmeyer, C., Pérez, A., Charron, N. E., De Fabritiis, G., Noé, F., and Clementi, C. Machine learning of coarse-grained molecular dynamics force fields. *ACS central science*, 5(5):755–767, 2019.

Wang, W. and Gómez-Bombarelli, R. Coarse-graining auto-encoders for molecular dynamics. *npj Computational Materials*, 5(1):1–9, 2019.

Wassenaar, T. A., Pluhackova, K., Bockmann, R. A., Marrink, S. J., and Tieleman, D. P. Going backward: a flexible geometric approach to reverse transformation from coarse grained to atomistic models. *Journal of chemical theory and computation*, 10(2):676–690, 2014.

Weiler, M. and Cesa, G. General e (2)-equivariant steerable cnns. In *Advances in Neural Information Processing Systems*, pp. 14334–14345, 2019.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. S. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Advances in Neural Information Processing Systems*, pp. 10381–10392, 2018.

Wilson, K. G. Renormalization group and critical phenomena. i. renormalization group and the kadanoff scaling picture. *Physical review B*, 4(9):3174, 1971.

Winkels, M. and Cohen, T. S. 3d g-cnns for pulmonary nodule detection. *arXiv preprint arXiv:1804.04656*, 2018.

Worrall, D. and Welling, M. Deep scale-spaces: Equivariance over scale. In *Advances in Neural Information Processing Systems*, pp. 7366–7378, 2019.

Xu*, M., Luo*, S., Bengio, Y., Peng, J., and Tang, J. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=pAbm1qfheGk.

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. *arXiv preprint arXiv:2105.07246*, 2021.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., and Smola, A. Deep sets. *arXiv preprint arXiv:1703.06114*, 2017.

Zhang, L., Han, J., Wang, H., Car, R., and E, W. Deepcg: Constructing coarse-grained models via deep neural networks. *The Journal of chemical physics*, 149(3):034101, 2018.

Zwanzig, R. *Nonequilibrium statistical mechanics*. Oxford university press, 2001.

Table 1: Summary of important notations.

Symbol	Meaning
	Preliminaries
$x \in \mathbb{R}^{n\times 3}$ and $X \in \mathbb{R}^{N\times 3}$	FG geometry and CG geometry
i/I	index over atoms/beads in FG/CG geometry
$M \in \mathbb{R}^{N\times n}$	CG projection operator
$M^+ \in \mathbb{R}^{n\times N}$	lift operator. $M^+M = \Pi_n$, $MM^+ = I_N$.
$Q \in \mathbb{R}^{3\times 3}$	3×3 orthogonal transformation $Qx = \{Qx_1, ...Qx_n\}$
$g \in \mathbb{R}^3$	translational vector. $x + g = \{x_1 + g, ..., x_n + g\}$
m	assignment map from $[n]$ to $[N]$
w_i	weights for atom i for CG projection
C_I	$C_I = (k \in [n] \vert m(k) = I)$
	Encoder and prior
$\mathbf{Enc}_\phi$ or $\mathbf{Enc}_\phi$(x,X)	encoder in VAE formulation. In CGVAE, it is instantiated as $q_\phi(z\vert x, X)$
$\mathbf{Prior}_\psi$ or $\mathbf{Prior}_\psi(X)$	prior in VAE formulation. In CGVAE, it is instantiated as $p_\psi(z\vert X)$.
Msg	message function
Update	update module
RBF : $\mathbb{R} \to \mathbb{R}^K$	radial basis function following Schütt et al. (2021); Klicpera et al. (2020)
$h_i^t/H_I^t \in \mathbb{R}^F$	FG/CG invariant embeddings at t_{th} convolution step in $\mathbf{Enc}_\phi$ or $\mathbf{Prior}_\psi$ model
$\widetilde{H}_I^t$	CG node embeddings updates for node I pooled from FG information in Eq. (4)
$\Delta H_I^t, \Delta V_I^t, \Delta \overline{H}_I^t, \Delta \overline{V}_I^t$	residual udpates for scalar, vector, pseudoscalar, and psuedovector at convolution depth t
$N(\cdot)$	the neighbor function to obtain adjacent node in a graph
F/K	dimension for node/edge features
$\mathcal{N}(\cdot\vert\cdot,\cdot)$	multivariate gaussian distribution
$z \in \mathbb{R}^{N\times F}$	latent representation on CG beads
ϵ	reparameterized Gaussian noise
$H^\phi \in \mathbb{R}^{N\times F}$	$H^\phi := \mathbf{Enc}_\phi(x, X) = p_\phi(z\vert x, X)$ which is the product of CG-wise normal distribution.
$H^\psi \in \mathbb{R}^{N\times F}$	$H^\psi := \mathbf{Prior}_\psi(X) = p_\psi(z\vert X)$ which is the product of CG-wise normal distribution.
	Decoder
$W_1 - W_9$	parameters for equivariant update in Eq. (5)
L_1, L_2	MLPs used to parameterize W in Eq. (5). $L_1 : \mathbb{R}^F \to \mathbb{R}^F$, $L_2 : \mathbb{R}^K \to \mathbb{R}^F$
Index(i, C_I)	index function. Index(i, C_I) retrieves the index of $i \in C_I$. For example, Index$(1, (1, 2, 4)) = 0$
Dec$(\cdot)$	a generic decoder function : $\mathbb{R}^{N\times 3} \to \mathbb{R}^{n\times 3}$
$\mathbf{Dec}_\theta$ or $\mathbf{Dec}_\theta(X, z)$	decoder in VAE formulation. In CGVAE, it is instantiated as $p_\theta(x\vert X, z)$.
$\widetilde{x}$	$\mathbb{R}^{n\times 3}$, decoded FG geometry. $\widetilde{x} = p_\theta(x\vert X, z)$.
$\Delta\widetilde{x}$/$\Delta\widetilde{x}_i$	$\Delta\widetilde{x} := \widetilde{x} - X$. $\Delta\widetilde{x}_i$ is i_{th} coordinate of $\Delta\widetilde{x}$. Note that decoder in CGVAE generates $\Delta\widetilde{x}$.
$H, V, \overline{H}, \overline{V}$	scalar, vector, pseudoscalar, pseudovector features
$V^\theta \in \mathbb{R}^{N\times F\times 3}$	equivariant output of decoder, subsets of its vector channels are used as prediction for $\Delta\widetilde{x}$
$\sigma(\cdot,\cdot)$	equivariant update function proposed in Schütt et al. (2021)
	Misc
$d_{\text{cut}}/D_{\text{cut}}$	cutoff distance to form a graph on FG/CG geometry
$\{\mathcal{G}, \mathcal{V}, \mathcal{E}\}$	graphs, nodes, and edges
$d_{ij}/d_{IJ} \in \mathbb{R}$	inter-node distances in FG/CG level.
d_{iI}	distance between i_{th} FG atom and I_{th} CG atom
$\widehat{E}_{IJ} = \frac{X_J - X_I}{d_{IJ}}$	unit distance vector between I and J
	Experiments
$\mathcal{G}_{\text{mol}}$	molecular graphs based on chemical bond connectivities
$\mathcal{G}_{\text{sample}}$	deduced molecular graphs of samples generated by CGVAE
$\mathcal{E}$	edge set that includes 2-hop neighbors in $\mathcal{G}_{\text{mol}}$
$\lambda(\cdot,\cdot)$	graph edit distance ratio to measure sample quality by comparing $\mathcal{G}_{\text{mol}}$ and $\mathcal{G}_{\text{sample}}$
$\text{RMSD}_{\text{recon}}$	measure reconstruction loss; the lower, the better
RMSD_{gen}	measure diversity of generated samples; the higher, the better.
	Operators
$\circ$	pointwise product
$[\cdot,\cdot]$	concatenation
$\times$	cross product

A. Symbol Table

We summarize definitions of important notations in Table 1.

B. Background and Related Works

B.1. MD Background

Molecular dynamics and CG. The dynamics and distribution of x are governed by a Hamiltonian $u(x)$. Under the ergodic hypothesis, the distribution of x converges to the Boltzmann distribution: $p(x) \propto e^{-u(x)\beta}$ where β is the inverse temperature of the system heat reservoir. Given m and $p(x)$, the equilibrium distribution of X can be derived by $\int p(x)p(X|x)\,dx$ where $p(X|x)$ is the dirac delta function $\delta(X - M(x))$ which map x deterministically onto the space of X. In statistical mechanics, $-\frac{1}{\beta}\ln p(X)$ is also defined as the Potential of Mean Force (PMF) which is a conservative free energy quantity to describe the equilibrium distributions of coarse-grained variables. In this work we are interested in solving the inverse problem of learning the generative map $p(x|X)$ given some mapping $M(x) = X$.

CG simulation. The theory and methodology of parameterizing CG models have been well established over decades (Kirkwood, 1935; Zwanzig, 2001; Noid et al., 2008). Systematic parameterization and mapping protocols have been widely applied to simulating large-scale biological molecules (Marrink et al., 2007; Menichetti et al., 2019; Souza et al., 2021). Lately, there are also a handful of works utilizing machine-learned potentials to simulate CG systems (Zhang et al., 2018; Wang & Gómez-Bombarelli, 2019; Wang et al., 2019; Husic et al., 2020).

B.2. More Related Works

Equivariant modeling of point sets. Efforts have been made to build equivariant neural networks for different types of data and symmetry groups (Zaheer et al., 2017; Qi et al., 2017; Maron et al., 2018; Thomas et al., 2018; Weiler et al., 2018; Cohen et al., 2019; Worrall & Welling, 2019; Weiler & Cesa, 2019; Cohen et al., 2019; Fuchs et al., 2020; Finzi et al., 2020; Maron et al., 2020; Bevilacqua et al., 2021) . In the case of 3D roto-translations for point clouds, tensor field network (TFN) (Thomas et al., 2018) and its extension with attention mechanism $SE(3)$ transformer (Fuchs et al., 2020) have been applied to various domain where modeling roto-translation is crucial (Winkels & Cohen, 2018; Unke et al., 2021; Batzner et al., 2021; Cai et al., 2021).

Generative model for molecular conformations. Many generative models have been proposed to generate molecular conformation ensembles conditioned on molecular graphs with modeling techniques like autoregressive models (Gebauer et al., 2019), conditional normalizing flow (Xu* et al., 2021), bilevel programming (Xu et al., 2021), and gradient-based optimizations (Shi et al., 2021).

C. Mathematical Details

C.1. Lift Operator

Definition C.1. *A lift operator M^+ of size $n \times N$ that maps the point sets space from $\mathbb{R}^{N\times 3}$ to $\mathbb{R}^{n\times 3}$ satisfies the following property.* $M^+_{i,I} = \begin{cases} 1 & \text{if } I = m(i) \\ 0 & \text{otherwise} \end{cases}$

In the case of $w_i = 1$ (w_i is the projection weight defined in Section 3.1) for all i, the M^+ is the Moore-Penrose pseudoinverse of M that has the following properties (Loukas, 2019; Cai et al., 2020).

Property C.1. $MM^+ = I_N$.

Proof. First note that $M^+_{i,I} = \mathrm{I}_{I\in m(i)}$ where I is the indicator function. Expanding the definition, we have

$$\sum_{i\in[n]} M_{I,i}M^+_{i,J} = \sum_{i\in[n]} \frac{w_i}{\sum_{j\in C_I} w_j}\mathrm{I}_{J\in m(i)} = \sum_{i\in[n]} \frac{w_i \mathrm{I}_{I\in m(i)}}{\sum_{j\in C_I} w_j}\delta_{IJ} = \frac{\sum_{i\in C_I} w_i}{\sum_{j\in C_I} w_j}\delta_{IJ} = \delta_{IJ}$$

□

C.2. A Toy Example

Consider a cluster map m that atoms $\{1,2,3,4\}$ into 2 groups $\{\{1,2,3\},\{4\}\}$. Assuming the mapping is constructed as geometric centers, i.e., the mapping weights are equal for each CG particle, then $M = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $M^+ = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$. It naturally follows that $MM^+ = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $M^+M = \Pi_n = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

Note that the normalization condition $\sum_i M[I,i] = 1$ is essential for M to respect translational equivariance because CG needs to preserve the norm of the translation vector g.

Note that the construction of CG projections do *not* require that $M_{I,i} \neq 0 \ \ \forall i \in C_I = \{k \in [n] | m(k) = I\}$ because w_i can take arbitrary non-negative values. Intuitively, this means that some atoms do not contribute to the construction of CG coordinates, but they are assigned to a bead via m in C_I. This can sometimes be seen in some coarse-grained simulations in practice: for instance protein simulations are often run with α carbons only. In the case where $w_i = 0$ for some i, M^+ the operator defined is no longer the pseudo-inverse of M. For example, if we assign 0 weights to two of the nodes in M induced by m is $M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Based on Definition C.1, because M^+ is constructed from m, M^+ remains the same but is not the pseudo-inverse of M in this case.

C.3. Proof of Property 3.1

Property 3.1. *Let $f_M : \mathbb{R}^{n\times 3} \to \mathbb{R}^{N\times 3}$ be the linear transformation $f_M(x) := Mx$. f_M is $E(3)$ equivariant, i.e., $f_M(Qx+g) = Qf_M(x) + g$, where Q is a 3×3 orthogonal matrix, and g is a translational vector.*[4]

Proof. Without loss of generality, we prove $f_M(Qx+g) = Qf_M(x) + g$ holds for any bead I.

$$[f_M(Qx+g)]_I = \sum_i M_{I,i}(Qx_i + g) = \sum_i M_{I,i}(Qx_i) + \sum_i M_{I,i}g = QX_I + \sum_i M_{I,i}g = QX_I + g = [Qf_M(x)+g]_I$$

The first equality follows from the definition of f_M. The second equality follows from the linearity of M. The third equality definition of Q and M. The fourth equality follows from the definition of M. The last equality follows from the definition of f_M. □

C.4. On the geometric Requirements for Backmapping

Given the fact that M is $E(3)$ equivariant, we can show that Dec has to be equivariant for **R1** to be true.

Proposition C.1. *If $M\text{Dec}(X) = X$ and M is $E(3)$ equivariant,* Dec *is also $E(3)$ equivariant, i.e.* Dec*(QX + g) = Q* Dec*(X) + g.*

Proof. We show $\text{Dec}(Qx+g) = Q\text{Dec}(X) + g$ by proof of contradiction. Assume $\text{Dec}(QX+g) \neq Q\text{Dec}(X) + g$. Given $M\text{Dec}(X) = X$ and M is $E(3)$ equivariant, we have

$$M\text{Dec}(QX+g) \neq MQ\text{Dec}(X) + Mg = MQ\text{Dec}(X) + g = QM\text{Dec}(X) + g = QX + g \tag{7}$$

The first inequality follows from the assumption. The second equality follows from the definition of M. The third equality follows from the property of M and Q. The last equality follows from the property of Dec. We reaches a contradiction that $M\text{Dec}(QX+g) \neq QX+g$. Therefore we conclude that the assumption has to be false and $\text{Dec}(QX+g) = Q\text{Dec}(X)+g$.

□

[4]Following Satorras et al. (2021), here Qx and $x+g$ are short-hands for $\{Qx_1, ..., Qx_n\}$ and $\{g+x_1, ..., g+x_n\}$. The same convention also applies to QX and $X+g$ at the coarse level.

C.5. ELBO Derivation

Following the probabilistic graphical model in Fig. 7, we carry out the standard derivation of the Evidence Lower Bound (ELBO) described in Eq. (1) for our conditional VAE here (Sohn et al., 2015):

$$\begin{aligned}
\log p(x|X) &= \log \mathbb{E}_{q_\phi(z|x,X)} \frac{p(x|X)}{q_\phi(z|x,X)} \\
&\geq \mathbb{E}_{q_\phi(z|x,X)} \log \frac{p(x|X)}{q_\phi(z|x,X)} \\
&= \mathbb{E}_{q_\phi(z|x,X)} \log \frac{p_\theta(x|X,z)p_\psi(z|X)}{q_\phi(z|x,X)} \\
&= \underbrace{\mathbb{E}_{q_\phi(z|x,X)} \log p_\theta(x|X,z)}_{\mathcal{L}_{\text{recon.}}} + \underbrace{\mathbb{E}_{q_\phi(z|x,X)} \log \frac{p_\psi(z|X)}{q_\phi(z|x,X)}}_{\mathcal{L}_{\text{reg.}}}
\end{aligned}$$

D. Graph Representations

Table 2: Summary of graph data inputs used in CGVAE.

	$\mathcal{G}_{\text{FG}}$	$\mathcal{G}_{\text{CG}}$
Coordinate	$x \in \mathbb{R}^{n\times 3}$	$X \in \mathbb{R}^{N\times 3}$
Node	$\mathcal{V}_{\text{FG}}$	$\mathcal{V}_{\text{CG}}$
Edge	$\mathcal{E}_{\text{FG}} = \{(i,j) \mid d(x_i,x_j) < d_{\text{cut}}, \forall\ i,j \in \mathcal{V}_{\text{FG}}\}$	$\mathcal{E}_{\text{CG}} = \{(I,J) \mid (C_I \times C_J) \cap \mathcal{E}_{\text{FG}} \neq \emptyset, \forall I,J \in \mathcal{V}_{\text{CG}}\}$
Node feature	invariant features $h \in \mathbb{R}^{n\times F}$	invariant features $H \in \mathbb{R}^{N\times F}$, equivariant features $V \in \mathbb{R}^{N\times F\times 3}$
Edge feature	$d_{ij} \in \mathbb{R}$	$d_{IJ} \in \mathbb{R}$, $\widehat{E}_{IJ} = \frac{X_I - X_J}{d_{IJ}} \in \mathbb{R}^3$

Here we describe the graph data structure we used for CG and FG graphs for geometric convolutions. The detailed description is summarized in Table 2.

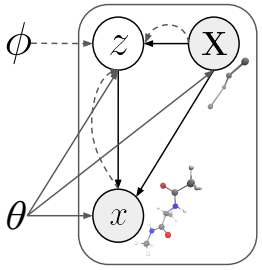

Figure 7: The probabilistic graphical model for the generation process (solid line) and inference process (dashed line) of the conditional geometric reverse mapping.

FG 3D graph $\mathcal{G}_{\text{FG}}$. The geometric graph at the FG level is represented as an undirected $\mathcal{G}_{\text{FG}} = (\mathcal{V}_{\text{FG}}, \mathcal{E}_{\text{FG}})$, where $\mathcal{V}_{\text{FG}} = [n]$ is the node set. The edge set is determined by a predefined distance cutoff d_{cut} that $\mathcal{E}_{FG} = \{(i,j) \mid d(x_i,x_j) < d_{\text{cut}}, \forall\ i,j \in \mathcal{V}\}$, where $d(a,b) = \|a-b\|_2$ is the euclidean distance between a and b. Following standard featurization procedure in machine learning techniques for molecular graph, the initial node features $h \in \mathbb{R}^F$ is simply one-hot encoding of atomic types. Following (Schütt et al., 2021), we parameterize edge features using $\text{RBF}(d_{ij})$ to expand d_{ij} with coefficients of K sine radial basis: $\sin(\frac{n\pi}{d_{\text{cut}}} d_{ij})/d_{ij}$ with a cosine filter as a continuous cutoff. n ranges from 1 to K. Both K and d_{cut} are hyper-parameters of the model.

CG graph. Given m, we introduce a corresponding undirected CG graph $\mathcal{G}_{\text{CG}}$ with $\mathcal{V}_{CG} = [N]$. There is an edge between node I and J if there exists at least an edge between sub-nodes in C_I and C_J: $\mathcal{E}_{\text{CG}} = \{(I,J) \mid (C_I \times C_J) \cap \mathcal{E}_{\text{FG}} \neq \emptyset, \forall I,J \in \mathcal{V}_{\text{CG}}\}$. Here, $C_I \times C_J$ indicates the Cartesian product between C_I and C_J. The initial node feature H_I is parameterized using the pooling operation described in Section 4.2. The distances for CG point sets are similarly featurized with sine radial basis functions, with a cutoff distance at D_{cut}. For the equivariant decoding described in Section 4.3, CG graph also carries a set of equivariant features parameterized from linear combinations of $\widehat{E}_{IJ}$.

FG bond graph. A molecule can have many conformations (stereo-isomers) which share the same bond graph structure. For classical MD simulations where there are no chemical reactions, the chemical bond graph does not change. The molecular graph $\mathcal{G}_{\text{mol}} = (\mathcal{V}_{\text{mol}}, \mathcal{E}_{\text{mol}})$ can be deduced from chemical rules and can be obtained from open-sourced chemoinformatic analysis packages like `RDKIT` (Landrum, 2013).

Determine bond graph from FG geometries. For evaluation of quality of generated samples, we compute the bond graphs of generated samples and compare them with the ground truth chemical bond graph. The chemical graphs are essentially radius graphs but with pairwise distance cutoffs determined by the atom types of the pairs involved. The distance cutoffs for

each pair of atoms are determined from the covalent radii of each atoms. We obtain covalent radius of each atom from `RDKIT` using `rdkit.Chem.PeriodicTable.GetRCovalent`. Let $r_{\text{covalent}}(i)$ indicates the covalent radius of atom i, the bond graph of generated geometries can be deduced: $\{(i,j) \mid \|x_i - x_j\|_2 < 1.3 \times (r_{\text{covalent}}(i) + r_{\text{covalent}}(j) \forall i,j \in \mathcal{V}_{\text{FG}}\})$. The scaling factor of 1.3 is chosen based on the `xyz2mol` public repository (Jensen, 2021).

E. Model Details

E.1. Details of Encoder and Prior Networks

Encoder. We introduce the detailed design of our encoder. Recall that K is the dimension of invariant edge features, and F is the dimension of node embedding. We define $L_3, L_3', L_3'', L_3''' : \mathbb{R}^K \to \mathbb{R}^F$ and $L_4, L_4', L_4'', L_4''' : \mathbb{R}^F \to \mathbb{R}^F$ be layer-wise MLPs with swish function as activation. The detailed form of FG MPNN is:

$$h_i^{t+1} = \text{Update}\left(\sum_{j\in N(i)} \text{Msg}(h_i^t, h_j^t, \text{RBF}(d_{ij})),\ h_i^t\right) = h_i^t + \sum_{j\in N(i)} L_3(\text{RBF}(d_{ij})) \circ L_4(h_i^t \circ h_j^t)$$

The detailed form of the pooling operation to map information from FG to CG level is:

$$\widetilde{H}_I^t = \text{Update}\left(\sum_{i\in C_{I=m(i)}} \text{Msg}(h_i^t, H_I^t, \text{RBF}(d_{iI})),\ H_I^t\right) = H_I^t + \sum_{i\in C_I} L_3'(\text{RBF}(d_{iI})) \circ L_4'(h_i^t)$$

Similar to MPNNs at the FG resolution, the CG-level MPNN is:

$$H_I^{t+1} = \text{Update}\left(\sum_{J\in N(I)} \text{Msg}(\widetilde{H}_I^t, \widetilde{H}_J^t, \text{RBF}(d_{IJ})),\ H_I^t\right) = H_I^t + \sum_{J\in N(I)} L_3''(\text{RBF}(d_{IJ})) \circ L_4''(\widetilde{H}_I^t \circ \widetilde{H}_J^t)$$

This form of MPNN is used for CG information propagation for both $\mathbf{Enc}_\phi$ and $\mathbf{Prior}_\psi$.

Prior. For $\mathbf{Prior}_\psi$, the detailed depth-wise form of the message passing operation is :

$$H_I^{t+1} = \text{Update}\left(\sum_{J\in N(I)} \text{Msg}(H_I^t, H_J^t, \text{RBF}(d_{IJ})),\ H_I^t\right) = H_I^t + \sum_{J\in N(I)} L_3'''(\text{RBF}(d_{IJ})) \circ L_4'''(H_I^t \circ \widetilde{H}_J^t)$$

E.2. Details of Decoder

In this subsection, We first give details about Eq. (5) and Eq. (6) in decoder and discuss the its equivariance property. We then discuss the geometric consistency of the decoder. Lastly, we give the implementation of channel selection operations.

E.2.1. Brief Introduction to Pseudoscalars and Pseudovectors

Pseudoscalars and pseudovectors are like scalars and vectors, but undergoes an additional sign flip under reflection. We use $\bar{\cdot}$ to indicate a quantity is pseudoscalar or psuedovector. Let $A, B, C \in \mathbb{R}^3$ be normal vectors. pseudoscalar $\overline{H}$ and pseudovector $\overline{V}$ can be constructed as follows: $\overline{H} = A \cdot (B \times C), \overline{V} = A \times B$. Note that cross product obeys the following identity under matrix transformations $(MA) \times (MB) = \det(M)(A \times B)$. Let P be a 3×3 reflection transformation with $\det(P) = -1$. It is easy to see that $(PA \times PB) = -P(A \times B)$. Along with the simple property of $PA \cdot PB = P(A \cdot B)$, we have $P\overline{H}$ and $P\overline{V}$

$$P\overline{H} = PA \cdot (PB \times PC) = PA \cdot (-P(B \times C)) = -A \cdot (B \times C) = -\overline{H}$$
$$P\overline{V} = PA \times PB = -P(A \times B) = -\overline{V}$$

Next, we list different ways to generate scalars/vectors/pseudoscalars/pseudovectors. These are used as the basis in Eq. (5). The details of these constructions for arbitrary order tensors can be found in the documentation of the e3nn package (Geiger et al., 2020).

- scalar: i) $\overline{H} \circ \overline{H}$ ii) $\overline{V} \cdot \overline{V}$
- vector: i) $V \times \overline{V}$ ii) $\overline{H} \circ \overline{V}$
- pseudoscalar: i) $V \cdot \overline{V}$ ii) $H \circ \overline{H}$
- pseudovector: i) $V \times V$ ii) $\overline{V} \times \overline{V}$ iii) $\overline{H} \circ V$

The output of Eq. (5) are scalars, pseudoscalars, vectors, and pseudovectors. As the construction of each equation in Eq. (5) ensures that the "type" of output matches with the "type" of the basis, we immediately get the equivariance guarantee for the outputs of Eq. (5).

E.2.2. Details of Equivariant Update Layers

In this subsection, we provide details of equivariant update layers in Eq. (6) $H_I^{t+1}, V_I^{t+1} = \sigma(H_I^t + \Delta H_I^t, V_I^t + \Delta V_I^t)$. Eq. (6) updates an invariant scalar feature $H_I \in \mathbb{R}^F$ as the first argument and an equivariant vector feature $V_I \in \mathbb{R}^{F\times 3}$ as the second argument. Under the action of rotation Q, σ has to satisfy the following constraint: $[\sigma(H_I, QV_I)]_1 = [\sigma(H_I, V_I)]_1$ and $[\sigma(H_I, QV_I)]_2 = Q[\sigma(H_I, V_I)]_2$. We next introduce the detailed forms of σ.

Let $L_5, L_5', L_5'' : \mathbb{R}^{2F} \to \mathbb{R}^F$ be MLPs and $W_\alpha, W_\beta \in \mathbb{R}^{F\times F}$ be two learnable matrices. Denote the invariant input of $\sigma(\cdot,\cdot)$ as $H_I'^{t+1} := H_I^t + \Delta H_I \in \mathbb{R}^F$ and the equivariant input as $V_I'^{t+1} := V_I^t + \Delta V_I \in \mathbb{R}^{F\times 3}$. The detailed form of σ to the update H^{t+1} and V^{t+1} is

$$\begin{aligned} H_I^{t+1} &= H_I'^{t+1} + L_5(\left[H_I'^{t+1}, \left\|W_\beta V_I'^{t+1}\right\|\right]) + L_5'(\left[H_I'^{t+1}, \left\|W_\beta V_I'^{t+1}\right\|\right])\langle W_\alpha V_I'^{t+1}, W_\beta V_I'^{t+1}\rangle \\ V_I^{t+1} &= V_I'^{t+1} + L_5''(\left[H_I'^{t+1}, \left\|W_\beta V_I'^{t+1}\right\|\right]) V_I'^{t+1} \end{aligned} \tag{8}$$

Here $\|\cdot\| : \mathbb{R}^{F\times 3} \to \mathbb{R}^F$ computes the norm for each feature channel. $\langle\cdot,\cdot\rangle : \mathbb{R}^{F\times 3} \times \mathbb{R}^{F\times 3} \to \mathbb{R}^F$ is the feature-wise dot product, and $[\cdot,\cdot] : \mathbb{R}^{F\times d_1} \times \mathbb{R}^{F\times d_2} \to \mathbb{R}^{F\times(d_1+d_2)}$ indicates the concatenation along the second feature channel. It is easy to show that the update function produces an invariant H_I^{t+1} and an equivariant V_I^{t+1} because 1) dot product and Euclidean norms are invariant under rotation and reflection, and 2) linear combinations of vectors are equivariant. More discussions can be found in Schütt et al. (2021).

E.2.3. Implementation of Channel Selection Operations

The final prediction of $\Delta\widetilde{x}$ are compiled from vector channels in $V_{I,\text{Index}(i,C_I)}$. This can be done efficiently by generating pairs of $(I, \text{Index}(i, C_I))$ used as indices array for advanced indexing in Pytorch. A PyTorch implementation is provided here:

```
def channel_select(V, m):
    # m: n x 1  CG map for each FG node
    # V : N x F x 3  vector channels on CG node
    channel_idx = torch.zeros_like( m)
    for cg_type in torch.unique(m):
        cg_filter = m == cg_type
        # find size of C_I
        k = cg_filter.sum().item() # find size of C_I
        # construct (I, Index(i, C_I))
        channel_idx[cg_filter] = torch.LongTensor(list(range(k)))
    dx = V[m, channel_idx]
    return dx
```

E.2.4. Ensuring Geometry Self-consistency

Just as the lift map M^+ satisfies $MM^+X = X$, the learnt decoder $\mathbf{Dec}_\theta$ also needs be compatible with pre-defined surjective CG map M, i.e. $M(\mathbf{Dec}_\theta(X, z)) = X$.

Proposition E.1. $\mathbf{Dec}_\theta(X, z) = M^+X + \Delta\widetilde{x} - M^+M\Delta\widetilde{x}$ *satisfies* $M(\mathbf{Dec}_\theta(X, z)) = X$.

Proof. From linearity of M and the property that $MM^+ = I$, we have

$$M(\mathbf{Dec}_\theta(X, z)) = MM^+X + M\Delta\widetilde{x} - MM^+M(\Delta\widetilde{x}) = X + M\Delta\widetilde{x} - M\Delta\widetilde{x} = X$$

□

F. Experiments Details

F.1. Baselines

Equivariant linear model. (Wang & Gómez-Bombarelli, 2019) We construct the first model following (Wang & Gómez-Bombarelli, 2019), the reconstructed coordinates are obtained from linear combinations of the coarse coordinates with learnable coefficients $D_{i,I}$. $\widetilde{x}_{i,m} := \sum_I D_{i,I} X_{I,m}$ where $m \in [3]$ indicates the Cartesian index of the coordinates. To incorporate translational equivariance which the model fails to incorporate, we recenter the FG structure and the mapped CG structure to its geometric center for training and testing of the model.

Table 3: Hyper-parameters for the MLP and equivariant linear baseline models. We used the same set of parameters for both alanine dipeptide and chignolin datasets.

	Alanine Dipeptide	Chignolin
$\mathcal{L}_{\text{graph}}$ weight γ	5	5
Training epochs	500	500
Learning rate	0.0001	0.0001
Batch size	32	32
Order for multi-hop graph	2	2

Multi-Layer Perceptrons.(An & Deshmukh, 2020) We use an MLP regressor as proposed by (An & Deshmukh, 2020) to map between CG space to FG space. After flattening the coordinate arrays, the MLP maps $\mathbb{R}^{3N}$ to $\mathbb{R}^{3n}$. The model does not incorporate $E(3)$ equivariance. We use MLPs with 2 hidden layers with hidden dimensions of size $3n$ with `ReLU` as the activation function.

Table 4: hyperparameters used for CGVAE

	Alanine Dipeptide	Chignolin
Dataset size	20000	10000
Edge feature dimension K	8	10
$\mathcal{L}_{\text{graph}}$ weight. γ	25.0	50.0
Encoder conv depth T^{enc}	4	2
Prior conv depth T^{prior}	4	2
Decoder conv depth T^{dec}	5	9
d_{cut}	8.5	12.0
D_{cut}	9.5	25.0
Node embedding dimension F	600	600
Batch size	32	2
Learning rate	8e-5	1e-4
Activation function	swish	swish
Training epochs	500	100
Order for multi-hop graph	2	2
β regularization strength of KL divergence	0.05	0.05

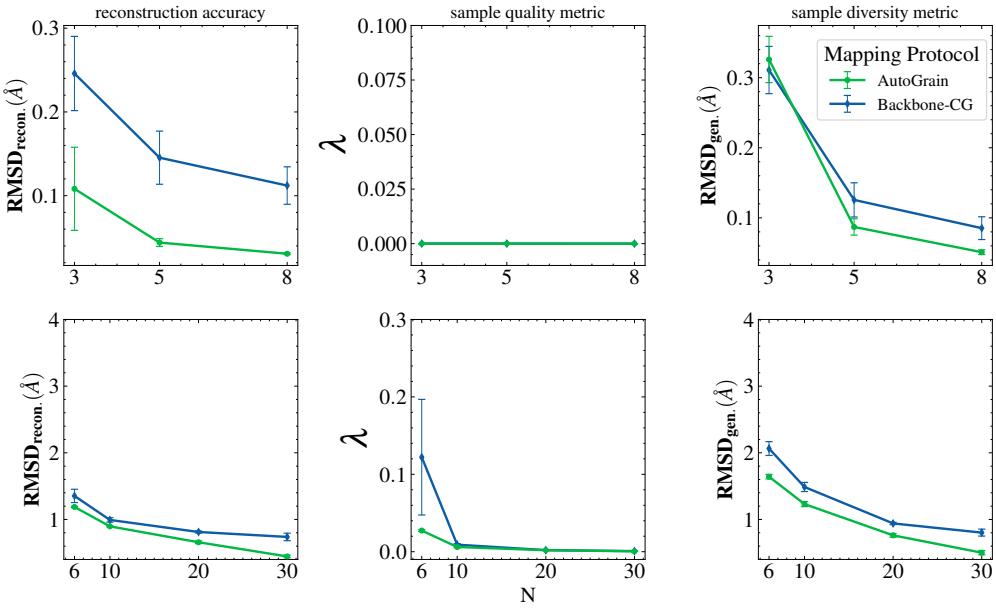

Figure 8: Comparing CGVAE results for alanine dipeptide (top) and chignolin (bottom) using two different mapping protocols. In general, the results from the two mappings are comparable, yet Auto-Grain yields better sample qualities for low-resolution Chignolin representations.

F.2. Details for CGVAE

Parameterize mappings with Auto-Grain. Our model works with arbitrary mapping protocols that satisfy the requirement described in Section 3.1. In practice, we find mappings that assign close atoms together perform better. We parameterize our CG mapping using Auto-Grain proposed in (Wang & Gómez-Bombarelli, 2019) which readers can refer for details of the method. Auto-Grain parameterizes the discrete assignment matrix $C_{I,i} \in \mathbb{R}^{N\times n}$ in a differentiable way using Gumbel-Softmax transformation (Jang et al., 2016) with a tunable fictitious temperature τ for different level of discreteness. The resulting CG transformation using the center of geometries is $M_{I,i} = \frac{C_{I,i}}{\sum_{i=1}^{n} C_{i,I}}$. A decoding matrix is used to equivariantly reconstruct x based on Mx with minimization of the reconstruction MSD loss. To encourage assignment of closed-together atoms into the same CG bead, we introduce a new geometric loss : $\mathcal{L}_{\text{geo}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{3}((C^T Mx)_{i,k} - x_{i,k})^2$ which penalizes mappings that groups atoms that are far away from each other. C^T can be thought as the lifting operator of the soft-parameterized M, satisfying $MC^T = I_N$. The total loss for Auto-Grain is $\mathcal{L}_{MSD} + a\mathcal{L}_{\text{geo}}$ where a re-weights the two loss contributions. We choose $a = 0.25$ in our experiments. For each experiment, we randomly select 1000 samples to train Auto-Grain to obtain a mapping. We train our model for 1500 epochs, with τ starting at 1.0 and gradually decreasing its value by 0.001 till it reaches a value of 0.025. The optimization is stochastic, and the obtained mapping will be slightly different, but compatible with CGVAE, for different parameter initializations. We find mapping obtained this way generate higher-quality samples than backbone-based CG protocol (introduced below), especially at low resolution for Chignolin. We present results obtained from CGVAE using the two mapping protocols in Fig. 8.

Hyperparameters for CGVAE. We use the hyperparameter set described in Table 4 for the CGVAE experiments we presented in the main text. The same set of hyperparameters are used for all choices of N. For optimization, we use the Adam optimizer (Kingma & Ba, 2014). We additionally apply a learning rate scheduler to adaptively reduce learning rate when convergence is reached. We use `torch.optim.lr_scheduler.ReduceLROnPlateau` implemented in Pytorch (Paszke et al., 2019) with `patience=15` and `factor=0.3`.

CG based on backbones. Performing CG transformation based on backbones is an alternative mapping strategy that groups neighboring atoms together. We randomly partition the linear backbone chain of the protein into N segments. We then assign the side chain atoms to the closest backbone segments based on the center of geometry of the backbone segments. We infer the backbone structure of the protein using the `MDTraj` package (McGibbon et al., 2015). After mapping is obtained, we compute X as the center of geometry.

F.3. Evaluation Metrics

Reconstruction accuracy. For the reconstruction task, we are interested in the difference between the original reference atomistic structure x and the reconstructed structure $\widetilde{x}$. This informs how well a model can reconstruct samples. For the training of CGVAEs, we remove the stochastic reparameterization to train a deterministic auto-encoders. This choice is to align our evaluation with our deterministic baselines. The discrepancy between x and $\widetilde{x}$ can be evaluated with root mean square distance $\text{RMSD}_{\text{recon}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \widetilde{x}_i)^2}{n}}$.

Sample qualities. The sampling task is performed based on the information of CG coordinates only. The invariant features $z \sim p_\psi(z|X)$ is then used to generate coordinates with the decoding function $\mathbf{Dec}_\theta(X, z)$. We are primarily interested in the quality of the generated geometries by evaluating how well the generated geometry ensembles preserve the bond graph $\mathcal{G}_{\text{gen}}$ compared with the graph $\mathcal{G}_{\text{mol.}}$ induced by the reference structure in the data. We infer $\mathcal{G}_{\text{gen}}$ and $\mathcal{G}_{\text{mol.}}$ with the protocol described in Appendix D. We quantify the similarity of the two generated graphs by computing the minimum graph edit distance. Because the generated geometries preserve the number of nodes and their identities, we simply need to compute the number of edge addition and removal operations required between $\mathcal{G}_{\text{gen}}$ and $\mathcal{G}_{\text{mol.}}$ and we denote $\mathcal{E}_{\text{gen}}$ and $\mathcal{E}_{\text{mol.}}$ as their edge sets respectively. Because the atom orders of the reference graph and the generated graphs are the same, the graph edit distance is the size of the symmetric difference (Δ), i.e, the union without the intersection of the two edge sets. We define the quantify $\lambda(\mathcal{E}_{\text{gen}}, \mathcal{E}_{\text{mol.}})$, the graph edit distance normalized by the size of the edge set of the original graph $\mathcal{E}_{\text{mol.}}$.

$$\lambda(\mathcal{E}_{\text{gen}}, \mathcal{E}_{\text{mol}}) = \frac{\text{GED}(\mathcal{E}_{\text{gen}}, \mathcal{E}_{\text{mol}})}{|\mathcal{E}_{\text{mol}}|} = \frac{\mathcal{E}_{\text{gen}} \Delta \mathcal{E}_{\text{mol}}}{|\mathcal{E}_{\text{mol}}|}$$

where Δ denotes the set difference. If $\lambda = 0$, it means that the molecular graph of the generated geometries is identical to the reference molecular graph. We evaluate λ for graphs for both the all-atom systems and heavy-atom only systems. We make the distinction because heavy-atoms encode most of the important dynamics of the system. Also, substructures that involve hydrogen are highly fluctuating, make the model (especially for models with few N) hard to capture its distribution fully. Additionally, we also compute the ratio of valid graphs of the generated samples. Such ratio tends to be low for Chignolin because of its more complex structures. For sample diversity metrics, we only calculate on structures with valid graphs. These values can be found in Table 5 and Table 6.

Sample diversities. We are also interested in the diversity of sampled geometries for our proposed generative model. Because M is a surjective map, each CG configuration corresponds to multiple possible configurations. The more diverse the sampled geometries given X, the more information that z has learned for the missing information encoded in z. For this purpose, we want to evaluate RMSD compared with the reference geometry to quantify how well the model can generate samples that are not in the reference set. The evaluation is only performed for generated samples with valid molecular graphs. The diversity RMSD is defined as: $\text{RMSD}_{\text{gen}} = \sqrt{\frac{\sum_i^n \int p_\psi(z|X)(x_i - \mathbf{Dec}(X,z))^2}{n}}$. Based on the discussion above, a higher value of RMSD_{gen} indicates better diversities of generated samples. To compute RMSD_{gen}, we sample 32 values of z for each X in the test set.

F.4. Tables and Sample Geometries

We show tabulated benchmark results in Table 5 for alanine dipeptide and chignolin datasets respectively for both heavy-atom and all-atom geometries. For RMSD_{gen}, the higher the value the better the model is in generating diverse samples. To complement the results of heavy atoms shown in Fig. 5 for heavy atoms, we also include plots for all-atom reconstruction and sampling in Fig. 9. We also compare generated chignolin samples ($N = 6$) from different methods in Fig. 10.

F.5. On Pseudoscalar Initialization for $N = 3$

As discussed in the main text, for $N = 3$, the vector basis sets are confined in a plane, so it is required to initialize pseudoscalar update with non-zero values to break the reflection symmetry. As it is shown in Fig. 11, the decoder with pseudoscalar initialized with 0_F produces coordinates that are confined in a plane, while a non-zero pseudoscalar initialization produces proper geometries that span the 3D space. For the case $N > 3$, it is less likely for CG beads to be in the same plane. However, when it is needed, one can always initialize pseudoscalar as non-zeros to break the symmetry even for larger N cases.

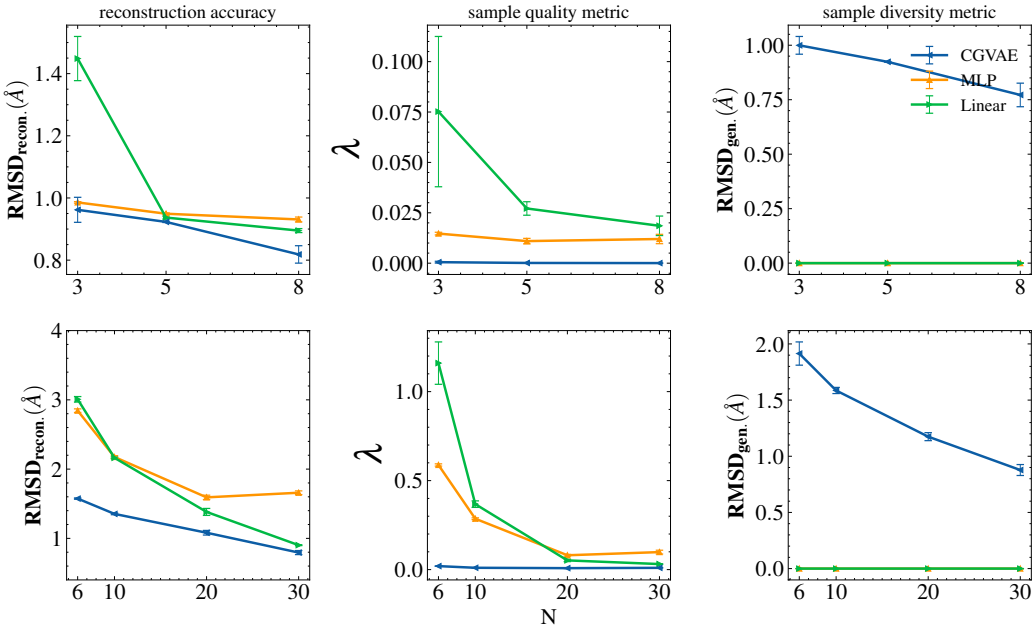

Figure 9: Benchmarks for all-atom reconstruction and sampled geometries at different resolutions for alanine dipeptide (top) and chignolin (bottom). RMSD_{gen} is not available for our deterministic baselines because the generation process is deterministic.

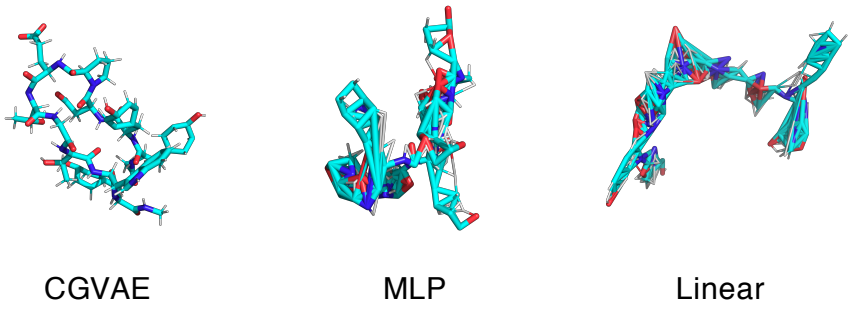

Figure 10: Comparison between generated chignolin samples ($N = 6$) from CGVAE and backmapped samples from baseline methods.

Table 5: Alanine dipeptide benchmarks

	Metric	Model	N		
			3	5	8
Heavy atom	**RMSD**$_{\text{recon}}$	linear	1.211 ± 0.075	0.277 ± 0.032	0.086 ± 0.009
		MLP	0.439 ± 0.003	0.288 ± 0.004	0.269 ± 0.009
		CGVAE	0.108 ± 0.050	0.044 ± 0.005	0.031 ± 0.001
	λ	linear	0.081 ± 0.045	0.001 ± 0.001	0.000 ± 0.000
		MLP	0.011 ± 0.001	0.001 ± 0.000	0.001 ± 0.000
		CGVAE	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	RMSD$_{\text{gen}}$	linear	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		MLP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		CGVAE	0.326 ± 0.033	0.087 ± 0.012	0.051 ± 0.003
	Valid graph ratio	linear	0.561 ± 0.153	0.990 ± 0.009	1.000 ± 0.000
		MLP	0.908 ± 0.011	0.992 ± 0.001	0.993 ± 0.002
		CGVAE	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
All atoms	**RMSD**$_{\text{recon}}$	linear	1.448 ± 0.071	0.937 ± 0.005	0.895 ± 0.006
		MLP	0.986 ± 0.002	0.949 ± 0.003	0.931 ± 0.008
		CGVAE	0.962 ± 0.040	0.923 ± 0.002	0.818 ± 0.028
	λ	linear	0.075 ± 0.037	0.027 ± 0.003	0.019 ± 0.005
		MLP	0.015 ± 0.001	0.011 ± 0.001	0.012 ± 0.002
		CGVAE	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	RMSD$_{\text{gen}}$	linear	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		MLP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		CGVAE	1.000 ± 0.041	0.924 ± 0.001	0.772 ± 0.054
	Valid graph ratio	linear	0.230 ± 0.151	0.667 ± 0.019	0.802 ± 0.048
		MLP	0.732 ± 0.015	0.808 ± 0.023	0.790 ± 0.038
		CGVAE	0.990 ± 0.010	0.998 ± 0.001	0.999 ± 0.000

Table 6: Chignolin benchmarks

	Metric	Model	N			
			6	10	20	30
heavy atom	**RMSD**$_{\text{recon}}$	linear	2.723 ± 0.055	1.845 ± 0.013	1.077 ± 0.036	0.702 ± 0.011
		MLP	2.558 ± 0.026	1.894 ± 0.013	1.318 ± 0.030	1.388 ± 0.024
		CGVAE	1.188 ± 0.010	0.897 ± 0.007	0.658 ± 0.020	0.445 ± 0.019
	λ	linear	1.124 ± 0.127	0.411 ± 0.028	0.085 ± 0.011	0.021 ± 0.002
		MLP	0.611 ± 0.009	0.331 ± 0.008	0.125 ± 0.008	0.146 ± 0.010
		CGVAE	0.027 ± 0.001	0.006 ± 0.001	0.002 ± 0.000	0.001 ± 0.000
	RMSD$_{\text{gen}}$	linear	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		MLP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		CGVAE	1.642 ± 0.034	1.231 ± 0.038	0.762 ± 0.028	0.502 ± 0.033
	valid graph ratio	linear	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.002	0.082 ± 0.022
		MLP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		CGVAE	0.415 ± 0.032	0.716 ± 0.034	0.843 ± 0.015	0.955 ± 0.012
all atom	**RMSD**$_{\text{recon}}$	linear	3.008 ± 0.040	2.163 ± 0.013	1.382 ± 0.050	0.902 ± 0.007
		MLP	2.844 ± 0.026	2.175 ± 0.020	1.592 ± 0.031	1.659 ± 0.027
		CGVAE	1.574 ± 0.009	1.353 ± 0.012	1.081 ± 0.033	0.795 ± 0.029
	λ	linear	1.160 ± 0.119	0.368 ± 0.018	0.052 ± 0.007	0.031 ± 0.002
		MLP	0.588 ± 0.006	0.286 ± 0.005	0.081 ± 0.005	0.098 ± 0.011
		CGVAE	0.020 ± 0.001	0.010 ± 0.001	0.008 ± 0.002	0.010 ± 0.000
	RMSD$_{\text{gen}}$	linear	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		MLP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		CGVAE	1.914 ± 0.103	1.586 ± 0.027	1.175 ± 0.036	0.877 ± 0.049
	valid graph ratio	linear	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		MLP	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
		CGVAE	0.072 ± 0.027	0.133 ± 0.024	0.278 ± 0.099	0.132 ± 0.011

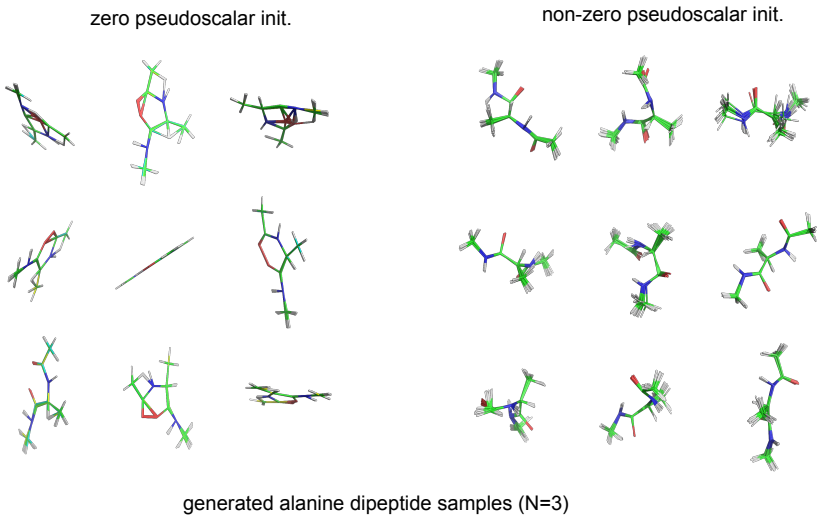

Figure 11: Comparing zero and non-zero pseudoscalar initializations for CGVAE at the resolution of $N = 3$ for alanine dipeptide.