Geometric Latent Diffusion Models for 3D Molecule Generation

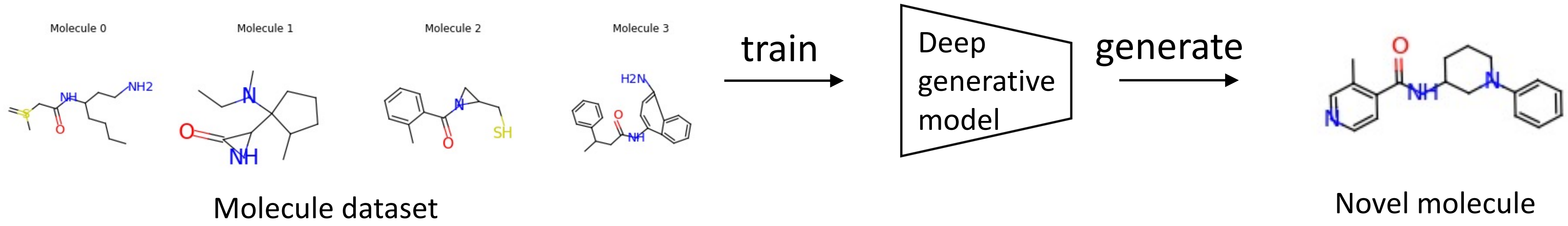
VIDY Reading Group Presentation

Presenter: Jialin Chen

Yale University

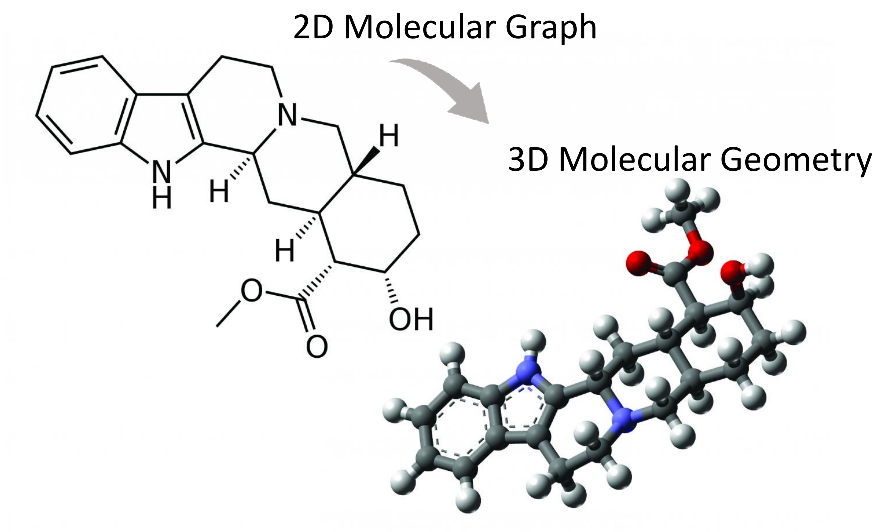
Molecule Generation

- Molecule generation task involves **creating novel molecules** with specific properties or structures, given a large real-world molecule dataset. The generated molecules should **meet desired criteria**, such as chemical validity, and stability.
- Recently, **deep generative models** have been widely applied to the problem of molecule generation.



3D Molecule Generation

- More and more molecule design frameworks and generation methods move the generation from **2D Molecular Graph** to **3D Molecular Geometry**.
- **Molecular Geometry** information is necessary for diverse downstream tasks like target drug design, protein design, antigen-specific antibody generation, etc.



Molecular Graph	node features (discrete / continuous) **edge index**
Molecular Geometry	node features (discrete / continuous) **3D coordinates**

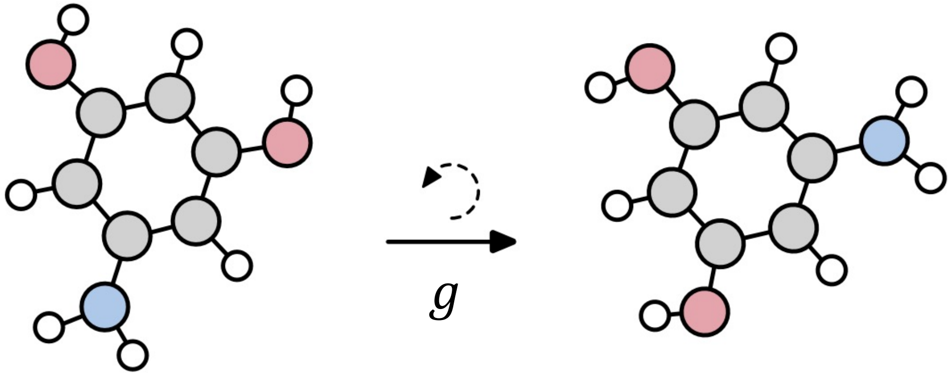
Task Definition

- **Unconditional Generation**: given a collection of molecules G, learn parameterized generative models $p_\theta(G)$ which can generate diverse and **realistic** molecules $\hat{G}$ in 3D.
- **Conditional Generation:** given a collection of molecules G with certain property s, learn **conditional** generative models $p_\theta(G|s)$ which can conditionally generate graph $\hat{G}$ which meets the property s.
  - Widely applied in drug discovery, antigen-specific antibody generation, etc.

Equivariance Constraint

The results of recent deep generative models (e.g., autoregressive and flow-based models) are still unsatisfactory with **low chemical validity**, due to the insufficient capacity of the underlying generative models.

- **Roto-translation Equivariance Constraint**
- $g = \boldsymbol{R} + \boldsymbol{t}$, for any translation $\boldsymbol{t}$ and orthogonal matrix $\boldsymbol{R}$

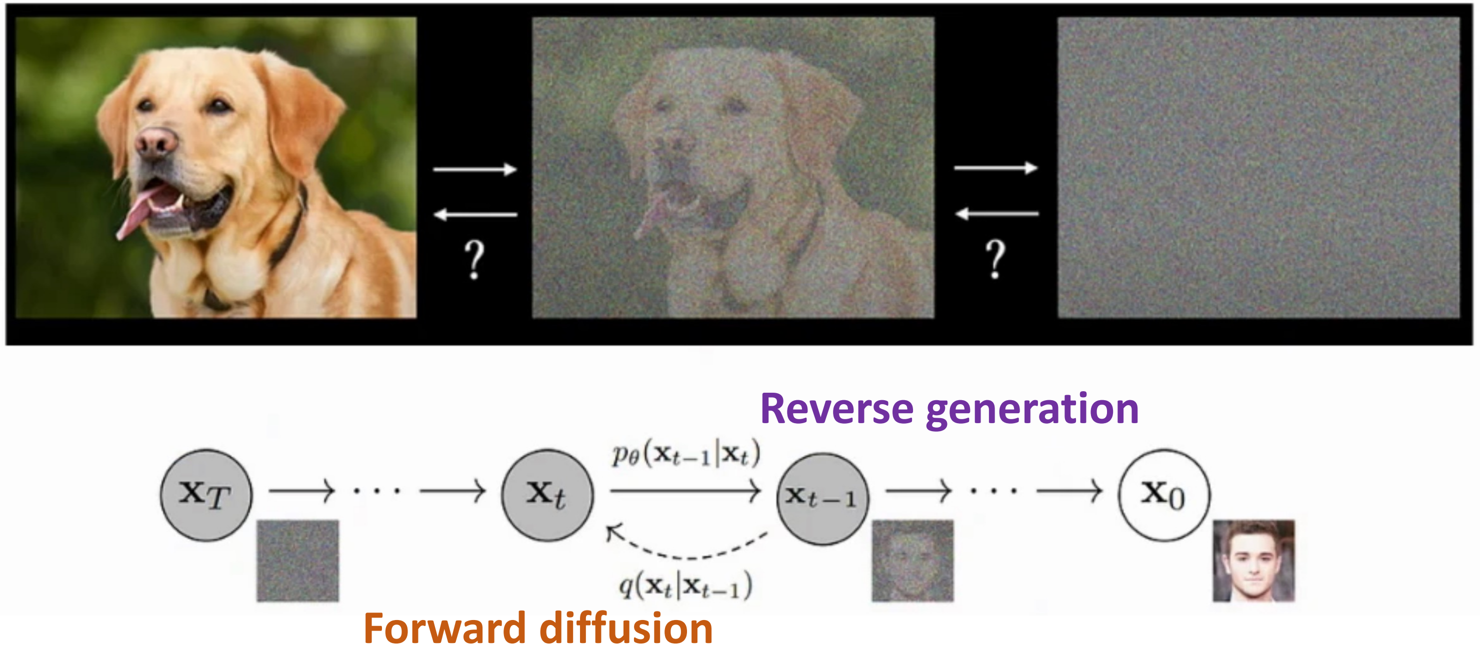


$$F \circ S_g(G) = T_g \circ F(G)$$

Note: the node features are intrinsically SE(3)-invariant, while the coordinates will be affected by SE(3) transformation

Previous Method: Diffusion Model

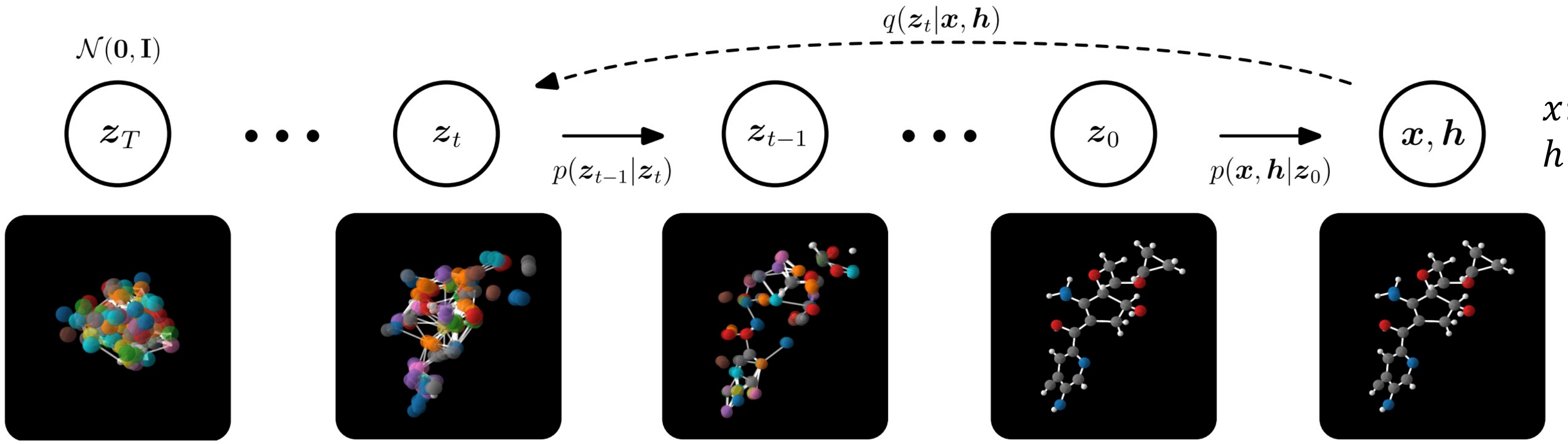
- Diffusion models (DMs) have emerged recently with surprising results on image tasks and beyond.
- Diffusion models define a **forward diffusion process** to destroy data into noise.
- Learn **reverse models** to generate realistic instances by denoising



Previous Method: Diffusion Model

Existing diffusion models (DMs) on molecules mainly work on the raw data space, which are

- multi-modal with discrete / integer / continuous variables, making unified diffusion process sub-optimal;
- high dimensional, making the model's training and sampling steps difficult.



x: 3D coordinates
h: node features

Resource: Equivariant Diffusion for Molecule Generation in 3D

Each sampling step constructs an intermediate graph in the original data space

Proposed Method

Geometric Latent Diffusion Model (GEOLDM) involves a variational autoencoder (AE) and a diffusion model (DM).

- The autoencoder (AE) learns a **smoother** and **lower-dimensional** latent space to embed molecular geometry and **unify diverse node features**
- The diffusion model (DM) operates on the latent space to implement **efficient training and sampling**
- GEOLDM utilizes equivariant networks as the forward functions to ensure 3D roto-translation equivariance ⇒ **satisfy chemistry validity**

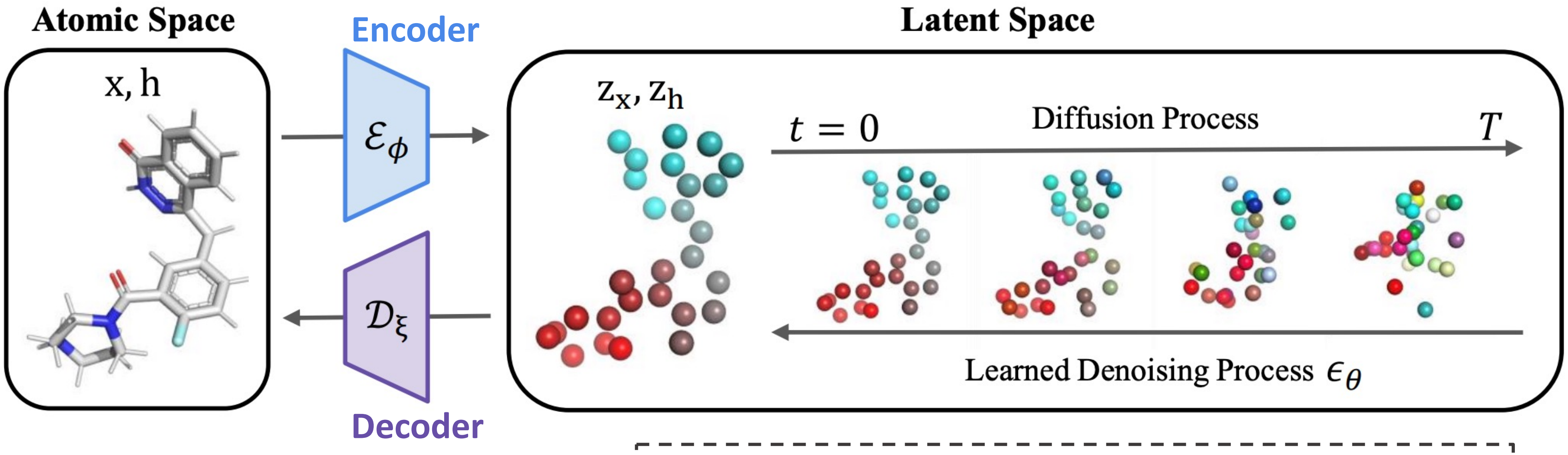
GEOLDM Pipeline

Each molecule is represented as $G(x, h)$; N is the number of nodes in the molecule.

$x = (x_1, x_2, \dots, x_N) \in \mathbb{R}^{N\times3}$ is the atom 3D coordinate matrix.

$h = (h_1, h_2, \dots, h_N) \in \mathbb{R}^{N\times d}$ is the node feature matrix, such as atomic type.

Latent representation $z_x \in \mathbb{R}^{N\times3}$ and $z_h \in \mathbb{R}^{N\times k}$ $(k < d)$



- Conduct denoising diffusion in the latent space instead of the data space
- Lead to better fidelity and controllable generation

Geometric Autoencoding (1)

- Encoder $\mathcal{E}_\phi$ encodes G into latent space: $(z_x, z_h) = \mathcal{E}_\phi(x, h) \in \mathbb{R}^{N\times(3+k)}$
  - In implementation: $(\mu_x, \mu_h) = \mathcal{E}_\phi(x, h)$; $(\epsilon_x, \epsilon_h) \sim \mathcal{N}(0, \boldsymbol{I})$
  - $(z_x, z_h) = (\epsilon_x \odot \sigma_0 + \mu_x, \epsilon_h \odot \sigma_0 + \mu_h)$
- Decoder D_ξ decodes back to the data space: $\tilde{G} = (\tilde{x}, \tilde{h}) = D_\xi(z_x, z_h)$

Theorem: Linear subspaces with the center always being zero can induce translation-invariant distribution.

- The latent z_x and reconstructed representation x are defined in the subspace that $\sum_i z_{x,i} = 0$ and $\sum_i x_i = 0$.

Geometric Autoencoding: EGNN (1)

- Both Encoder and Decoder are parameterized with EGNN architecture
- EGNNs are a class of Graph Neural Network that satisfies the equivariance property.
- In EGNN, molecular geometries are considered as point clouds (i.e., fully connected graphs), without specifying the connecting bonds.
- Each node v_i is embedded with coordinates $x_i \in \mathbb{R}^3$ and atomic features $h_i \in \mathbb{R}^d$.

Geometric Autoencoding: EGNN (2)

- Equivariant Graph Convolution Layer: $(x^{l+1}, h^{l+1}) = \mathrm{EGCL}(x^l, h^l)$

$$m_{ij} = \phi_e(h_i^l, h_j^l, d_{ij}^2, a_{ij}), \qquad h_i^{l+1} = \phi_h\left(h_i^l, \sum_{j\neq i} \tilde{e}_{ij} m_{ij}\right),$$

$$x_i^{l+1} = x_i^l + \sum_{j\neq i} \frac{x_i^l - x_j^l}{\boxed{d_{ij}+1}} \phi_x(h_i^l, h_j^l, d_{ij}^2, a_{ij}).$$

Normalize to improve the model stability

- $d_{ij} = \|x_i - x_j\|_2$, which is invariant to rotations. a_{ij} are optional edge features
- $\tilde{e}_{ij}$ represents the attention value between node i and node j
- All learnable functions ϕ_e, ϕ_h, ϕ_x are parameterized by MLPs
- Rotation Equivariance Principle: $Rx_i - Rx_j = R(x_i - x_j)$

Geometric Autoencoding (2)

- Since $\mathcal{E}_\phi$ and D_ξ are parameterized with equivariant graph neural networks (EGNNs), therefore

$$(\boldsymbol{R}z_x, z_h) = \mathcal{E}_\phi(\boldsymbol{R}x, h)$$

$$(\boldsymbol{R}x, h) = D_\xi(\boldsymbol{R}z_x, z_h)$$

- Overall, $(\boldsymbol{R}x + \boldsymbol{t}, h) = D_\xi(\mathcal{E}_\phi(\boldsymbol{R}x + \boldsymbol{t}, h))$ for all rotations $\boldsymbol{R}$ and translations $\boldsymbol{t}$

Loss Function for AE

- Loss function for Autoencoder (AE):

$$L_{AE} = L_{recon} + L_{reg}$$

- Reconstruction loss $L_{recon} = -\mathbb{E}_{q_\phi(z_x, z_h|x,h)} p_\xi(x, h|z_x, z_h)$ maximizes the likelihood of the data x, h
  - L_2 norm for continuous features and cross-entropy loss for discrete features
- Regularization loss L_{reg} penalizes $q_\phi(z_x, z_h|x, h)$ towards standard Gaussian to prevent latent embeddings from arbitrarily high variance

Geometric Latent Diffusion Models (1)

- Instead of conducting diffusion process on the initial graph (x, h), GEOLDM conducts on the latent variable $z = (z_x, z_h)$
- Forward process:

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t} z_{t-1}, \beta_t \boldsymbol{I})$$

$$q(z_{1:T}|z_0) = \prod_{t=1}^{T} q(z_t|z_{t-1})$$

- The final state z_T approximately converges to standard Gaussian distribution $\mathcal{N}(0, \boldsymbol{I})$

Geometric Latent Diffusion Models (2)

- Since $z_t = \alpha_t z_0 + \sigma_t \epsilon$ with $\alpha_t = \sqrt{\prod_{s=1}^{t}(1-\beta_s)}$ and $\sigma_t = \sqrt{1-\alpha_t^2}$, where $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$, the target of the denoising model ϵ_θ is to reconstruct ϵ.
- Loss function for the diffusion model:

$$L_{DM} = -\mathbb{E}_{z_0, \epsilon \sim \mathcal{N}(0,\boldsymbol{I}), t}[w(t)\|\epsilon - \epsilon_\theta(z_t, t)\|^2]$$

- The reweighting terms $w(t) = \frac{\beta_t^2}{2\rho_t^2(1-\beta_t)(1-\alpha_t^2)}$
- The reverse step:

$$z_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(z_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}}\epsilon_\theta(z_t, t)\right) + \rho_t \epsilon$$

Geometric Latent Diffusion Models (2)

Theorem: If the initial distribution $p(z_{x,T}, z_{h,T})$ is invariant and the transitions $p_\theta(z_{x,t-1}, z_{h,t-1} | z_{x,t}, z_{h,t})$ are equivariant, i.e., $p_\theta(z_{x,t-1}, z_{h,t-1} | z_{x,t}, z_{h,t}) = p_\theta(\boldsymbol{R}z_{x,t-1}, z_{h,t-1} | \boldsymbol{R}z_{x,t}, z_{h,t})$, then $p_\theta(z_x, z_h)$ is invariant: $p_\theta(z_x, z_h) = p_\theta(\boldsymbol{R}z_x, z_h)$

- To ensure the Roto-translation equivariance, the denoising network ϵ_θ is also parameterized by equivariance networks:

$$\left(\boldsymbol{R}z_{x,t-1} + \boldsymbol{t}, z_{h,t-1}\right) = \epsilon_\theta\left(\boldsymbol{R}z_{x,t} + \boldsymbol{t}, z_{h,t}, t\right)$$

Also subtract center of gravity from all the intermediate states $z_{x,t}$ to ensure translation invariance

  - In this way, $p_\theta(z_{x,t-1}, z_{h,t-1} | z_{x,t}, z_{h,t})$ is equivariant to rotation transformation (see paper for detailed proof)

Pseudo Algorithms

Algorithm 1 Training Algorithm of GeoLDM

1: **Input:** geometric data $\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle$
2: **Initial:** encoder network $\mathcal{E}_\phi$, decoder network $\mathcal{D}_\xi$, denoising network $\boldsymbol{\epsilon}_\theta$
3: **First Stage: Autoencoder Training**
4: **while** ϕ, ξ have not converged **do**
5: $\quad \boldsymbol{\mu}_{\mathrm{x}}, \boldsymbol{\mu}_{\mathrm{h}} \leftarrow \mathcal{E}_\phi(\mathbf{x}, \mathbf{h})$ {Encoding}
6: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
7: $\quad$ Subtract center of gravity from $\boldsymbol{\epsilon}_{\mathrm{x}}$ in $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_{\mathrm{x}}, \boldsymbol{\epsilon}_{\mathrm{h}}]$
8: $\quad \mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}} \leftarrow \boldsymbol{\epsilon} \odot \sigma_0 + \boldsymbol{\mu}$ {Reparameterization}
9: $\quad \tilde{\mathbf{x}}, \tilde{\mathbf{h}} \leftarrow \mathcal{D}_\xi(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})$ {Decoding}
10: $\quad \mathcal{L}_{AE} = \text{reconstrcution}([\tilde{\mathbf{x}}, \tilde{\mathbf{h}}], [\mathbf{x}, \mathbf{h}]) + \mathcal{L}_{reg}$
11: $\quad \phi, \xi \leftarrow \text{optimizer}(\mathcal{L}_{AE}; \phi, \xi)$
12: **end while**
13: **Second Stage: Latent Diffusion Models Training**
14: Fix encoder parameters ϕ
15: **while** θ have not converged **do**
16: $\quad \mathbf{z}_{\mathrm{x},0}, \mathbf{z}_{\mathrm{h},0} \sim q_\phi(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}} | \mathbf{x}, \mathbf{h})$ {As lines 5-8}
17: $\quad t \sim \mathrm{U}(0, T)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
18: $\quad$ Subtract center of gravity from $\boldsymbol{\epsilon}_{\mathrm{x}}$ in $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_{\mathrm{x}}, \boldsymbol{\epsilon}_{\mathrm{h}}]$
19: $\quad \mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t} = \alpha_t[\mathbf{z}_{\mathrm{x},0}, \mathbf{z}_{\mathrm{h},0}] + \sigma_t \boldsymbol{\epsilon}$
20: $\quad \mathcal{L}_{LDM} = ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}, t)||^2$
21: $\quad \theta \leftarrow \text{optimizer}(\mathcal{L}_{LDM}; \theta)$
22: **end while**
23: **return** $\mathcal{E}_\phi, \mathcal{D}_\xi, \boldsymbol{\epsilon}_\theta$

First count the distribution $p(N)$ of molecular sizes N on the training set. Then sample $N \sim p(N)$ and generate latent variables and node features in size N

Algorithm 2 Sampling Algorithm of GeoLDM

Input: decoder network $\mathcal{D}_\xi$, denoising network $\boldsymbol{\epsilon}_\theta$
$\mathbf{z}_{\mathrm{x},T}, \mathbf{z}_{\mathrm{h},T} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for t in $T, T-1, \cdots, 1$ **do**
$\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ {Latent Denoising Loop}
$\quad$ Subtract center of gravity from $\boldsymbol{\epsilon}_{\mathrm{x}}$ in $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_{\mathrm{x}}, \boldsymbol{\epsilon}_{\mathrm{h}}]$
$\quad \mathbf{z}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t^2}} \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)) + \rho_t \boldsymbol{\epsilon}$
end for
$\mathbf{x}, \mathbf{h} \sim p_\xi(\mathbf{x}, \mathbf{h} | \mathbf{z}_{\mathrm{x},0}, \mathbf{z}_{\mathrm{h},0})$ {Decoding}
return $\mathbf{x}, \mathbf{h}$

Experiment: Setup (1)

- Task: Molecular Modeling and Generation; Controllable Molecule Generation
- Dataset:
  - QM9 contains 3D structures together with several quantum properties for 130k small molecules, limited to 9 heavy atoms.
  - DRUG dataset consists of much larger organic compounds, with up to 181 atoms in 5 different atom types.

Experiment: Setup (2)

Evaluation:

- Given the generated molecular geometries (x, h), first predict bond types (single, double, triple, or none) by pair-wise atomic distances and atom types.
- Train on the training set and generate 10, 000 samples for evaluation
- Atom stability (%): the proportion of atoms that have the right valency
- Molecule stability (%): the proportion of generated molecules for which all atoms are stable.
- validity and uniqueness: measured by RDKIT.

Experiment: Results

ENF: equivariant generative model with normalization flow
G-Schnet: equivariant generative model with autoregressive model
GDM: graph diffusion model without equivariance
EDM: equivariant graph diffusion model on the raw data space
-AUG: train with data augmented by random rotations
GraphLDM: ablation of GEOLDM with only invariant latent variables

Table 1. Results of atom stability, molecule stability, validity, and validity×uniqueness. A higher number indicates a better generation quality. Metrics are calculated with 10000 samples generated from each model. On QM9, we run the evaluation for 3 times and report the derivation. Note that, for DRUG dataset, molecule stability and uniqueness metric are omitted since they are nearly 0% and 100% respectively for all the methods. Compared with previous methods, the latent space with both invariant and equivariant variables enables GEOLDM to achieve up to 7% improvement for the validity of large molecule generation.

# Metrics	QM9 Atom Sta (%)	QM9 Mol Sta (%)	QM9 Valid (%)	QM9 Valid & Unique (%)	DRUG Atom Sta (%)	DRUG Valid (%)
Data	99.0	95.2	97.7	97.7	86.5	99.9
ENF	85.0	4.9	40.2	39.4	-	-
G-Schnet	95.7	68.1	85.5	80.3	-	-
GDM	97.0	63.2	-	-	75.0	90.8
GDM-AUG	97.6	71.6	90.4	89.5	77.7	91.8
EDM	98.7	82.0	91.9	90.7	81.3	92.6
EDM-Bridge	98.8	84.6	92.0*	90.7	82.4	92.8*
GRAPHLDM	97.2	70.5	83.6	82.7	76.2	97.2
GRAPHLDM-AUG	97.9	78.7	90.5	89.5	79.6	98.0
GEOLDM	**98.9** $\pm$ 0.1	**89.4** $\pm$ 0.5	**93.8** $\pm$ 0.4	**92.7** $\pm$ 0.5	**84.4**	**99.3**

Experiment: Results

- **Diffusion-based** generative models outperform other generative models in the generation task of molecular geometries.
- **Data augmentation** boosts the quality of the generated samples.
- **Equivariance constraints** play a vital role in improving the performance.
- **GEOLDM** beats all baselines and achieve **SOTA** performance on two datasets.

Table 1. Results of atom stability, molecule stability, validity, and validity×uniqueness. A higher number indicates a better generation quality. Metrics are calculated with 10000 samples generated from each model. On QM9, we run the evaluation for 3 times and report the derivation. Note that, for DRUG dataset, molecule stability and uniqueness metric are omitted since they are nearly 0% and 100% respectively for all the methods. Compared with previous methods, the latent space with both invariant and equivariant variables enables GEOLDM to achieve up to 7% improvement for the validity of large molecule generation.

	QM9				DRUG	
# Metrics	Atom Sta (%)	Mol Sta (%)	Valid (%)	Valid & Unique (%)	Atom Sta (%)	Valid (%)
Data	99.0	95.2	97.7	97.7	86.5	99.9
ENF	85.0	4.9	40.2	39.4	-	-
G-Schnet	95.7	68.1	85.5	80.3	-	-
GDM	97.0	63.2	-	-	75.0	90.8
GDM-AUG	97.6	71.6	90.4	89.5	77.7	91.8
EDM	98.7	82.0	91.9	90.7	81.3	92.6
EDM-Bridge	98.8	84.6	92.0*	90.7	82.4	92.8*
GRAPHLDM	97.2	70.5	83.6	82.7	76.2	97.2
GRAPHLDM-AUG	97.9	78.7	90.5	89.5	79.6	98.0
GEOLDM	**98.9** $\pm$ 0.1	**89.4** $\pm$ 0.5	**93.8** $\pm$ 0.4	**92.7** $\pm$ 0.5	**84.4**	**99.3**

Experiment: Conditional Generation (1)

- Pipeline:
  - Split the training set of QM9 dataset into two halves (50K samples each)
  - Train a property predictor w on the first half and train a conditional model f on the second half.
  - given a property value s, conditionally draw samples from the generative model f and then use w to calculate their property values as $\hat{s}$
- Metric: MSE error between s and $\hat{s}$. lower, better.

Experiment: Conditional Generation (2)

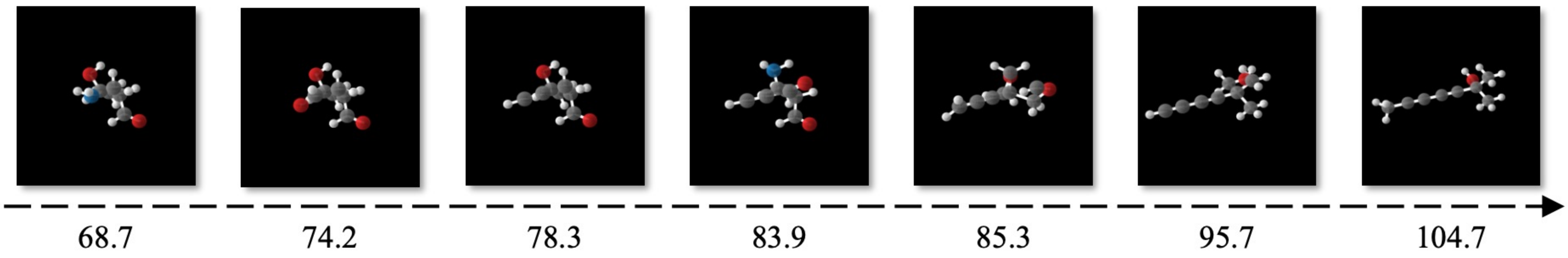
- **QM9**: directly run the property predictor w on the second half => indicate the bias of w. A smaller gap with QM9 numbers indicates a better property-conditioning performance.
- **Random**: randomly shuffle the property labels in the dataset and then evaluate w on it. This operation removes any relation between molecule and property.
- **Natoms**: predict the molecular properties by only using the number of atoms
- The results of QM9 and Random can be viewed as lower and upper bounds of MAE.

Table 2. Mean Absolute Error for molecular property prediction. A lower number indicates a better controllable generation result. Results are predicted by a pretrained EGNN classifier ω on molecular samples extracted from individual methods.

Property	α	$\Delta\varepsilon$	$\varepsilon_{\mathrm{HOMO}}$	$\varepsilon_{\mathrm{LUMO}}$	μ	C_v
Units	Bohr^3	meV	meV	meV	D	$\frac{\mathrm{cal}}{\mathrm{mol}}$K
QM9*	0.10	64	39	36	0.043	0.040
Random*	9.01	1470	645	1457	1.616	6.857
N_{atoms}	3.86	866	426	813	1.053	1.971
EDM	2.76	655	356	584	1.111	1.101
GEOLDM	2.37	587	340	522	1.108	1.025

Experiment: Conditional Generation (2)

Different **Polarizability values** α while keeping the reparameterization noise ε fixed. Typically, **less isometrically** molecular geometries lead to larger α values



Conclusion

- GEOLDM: a novel latent diffusion model for molecular geometry generation.
- GEOLDM overcomes the limitations of current molecular generative models by learning **diffusion models** over a **continuous, lower-dimensional latent** space with rotation and translation equivariance.
- GEOLDM builds **point-structured** latent encodings with both **invariant scalars** and **equivariant tensors.**
- Experimental results demonstrate its significantly better capacity for modeling chemically realistic molecules and controllable generation of molecular geometries.