E(n) Equivariant Normalizing Flows

35th Conference on Neural Information Processing Systems (**NeurIPS 2021**)

Victor Garcia Satorras[1*], *Emiel Hoogeboom*[1*], *Fabian B. Fuchs*[2],
Ingmar Posner[2], *Max Welling*[1]

UvA – Bosch Delta Lab, University of Amsterdam[1],
Department of Engineering Science, University of Oxford[2]

Reporter: Fanmeng Wang
May 12, 2022

Outline

- Introduction
- Background
- E(n) Equivariant Normalizing Flows
- Experiments
- Conclusions

Introduction

This paper focus on symmetries of the **n-dimensional Euclidean group**, referred to as E(n).

Propose a generative model equivariant to Euclidean symmetries:

E(n) Equivariant Normalizing Flows (E-NFs)

Motivation: To generate better drug candidate molecules to speed drug discovery

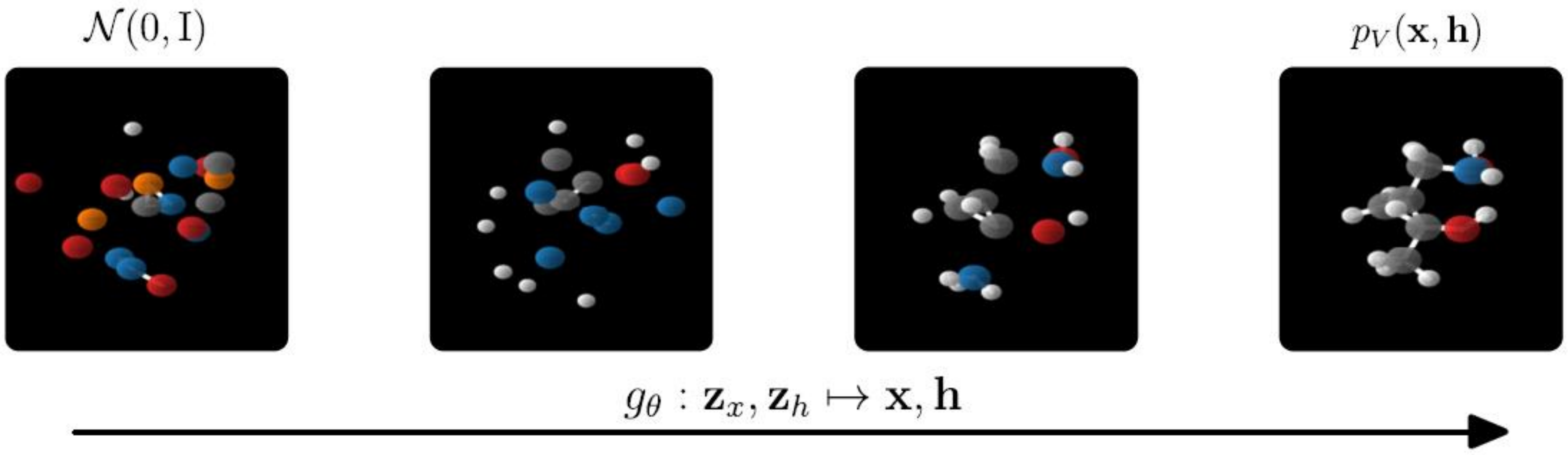
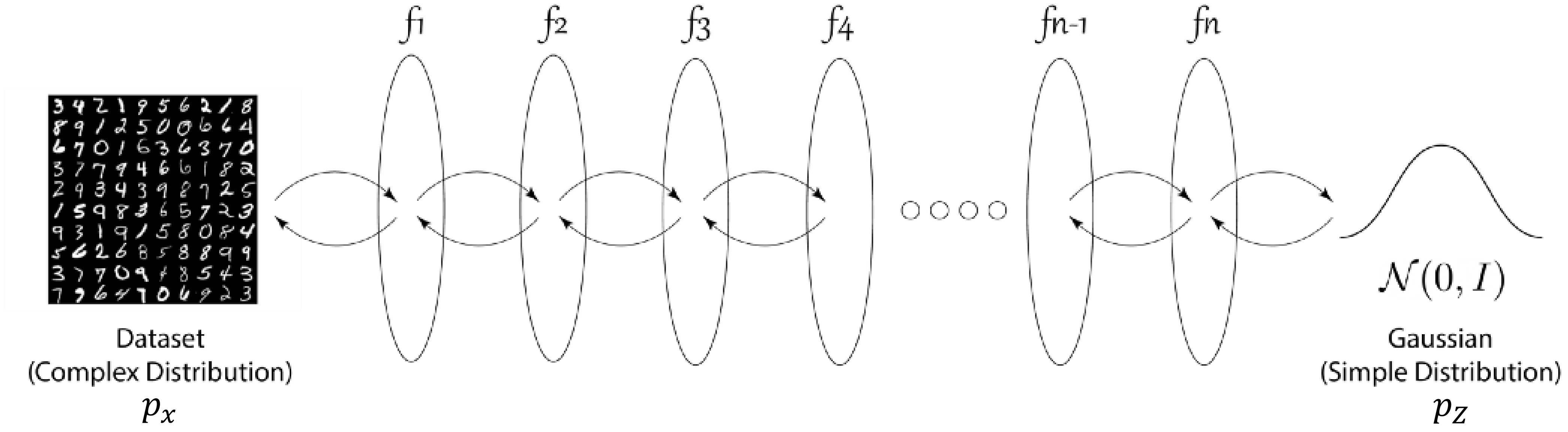


Figure 1: Overview of our method in the sampling direction. An equivariant invertible function g_θ has learned to map samples from a Gaussian distribution to molecules in 3D, described by $\mathbf{x}, \mathbf{h}$.

Background | E(n) Equivariant Normalizing Flows



- A complex distribution p_x
- A simple base distribution p_Z (such as a Gaussian distribution)
- A **learnable invertible transformation** f **:**

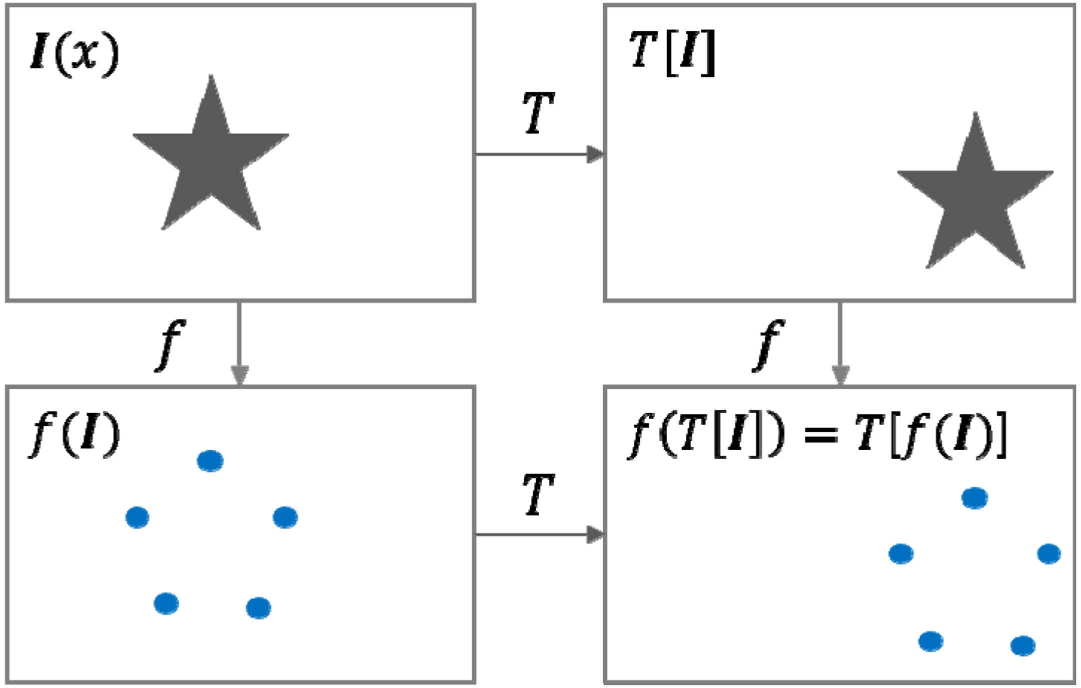
$$x = g(z)$$
$$z = g^{-1}(x) = f(x)$$

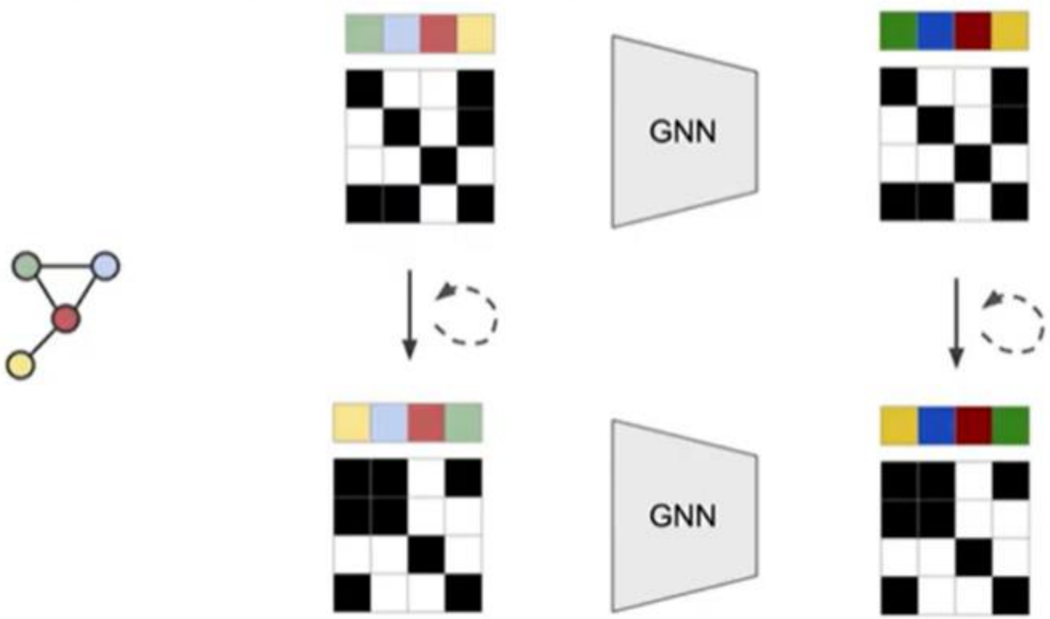
$$\rightarrow$$

$$p_x(x) = p_Z(z)\, |\det J_f(x)|$$
$$\log p_x(x) = \log p_Z(z) + \log |\det J_f(x)|$$

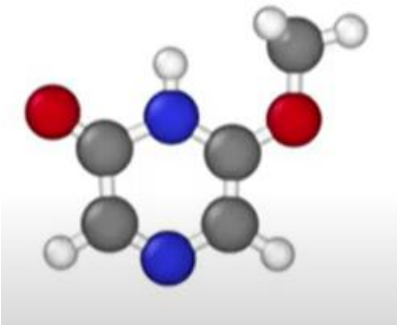
Equivariance of a function f: Transforming its input results in an equivalent transformations of its output.

$$f(T[I]) = T[f(I)]$$





GNN is built to be equivalent to permutations: Although we permute the indexes of nodes, it still represent the same graph.



E(n) Equivariant data such as molecules in 3D:

A molecular graph $G = (V, E)$ with nodes $v_i \epsilon V$ and edges e_{ij} .

Each node v_i is associated with a **position vector** x_i and **node features** h_i.

Features h have the property that they are invariant to E(n) transformations, while they do affect x.

In other words, **rotations and translations of $\boldsymbol{x}$ do not influence $\boldsymbol{h}$.**

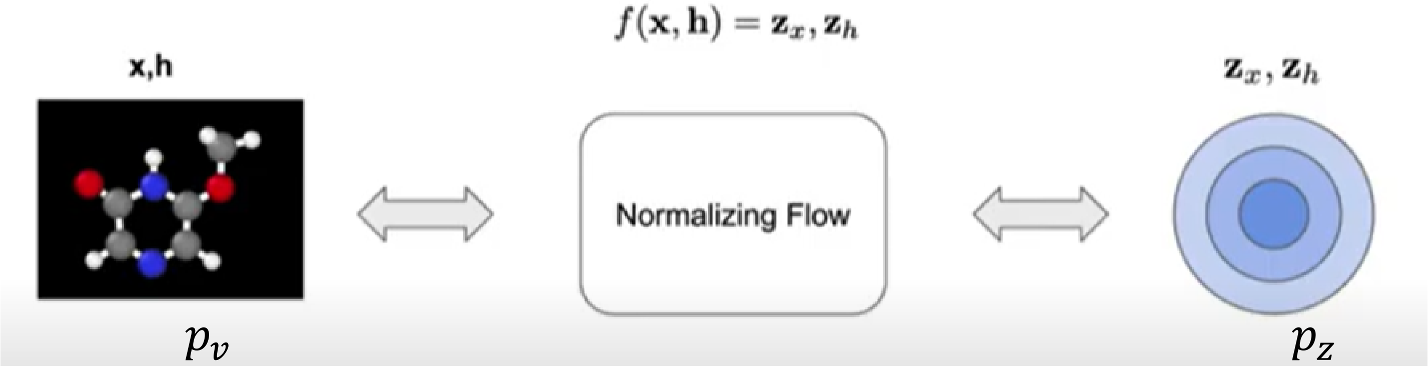
E(n) Equivariant Graph Neural Networks (EGNNs) （Satorras et al. ICML2021）

EGNNs is **equivariant to Euclidean symmetries** (rotation, translation, reflection and displacement et al） .

However, this model is only able to discriminate features on nodes, and cannot generate new graphs.

E(n) Equivariant Normalizing Flows

A probabilistic model for data with Euclidean symmetry.



$V = (x, h)$

- position vector $x \in R^{M*n}$ embedded in a n-dimensional space
- node features $h \in R^{M*nf}$

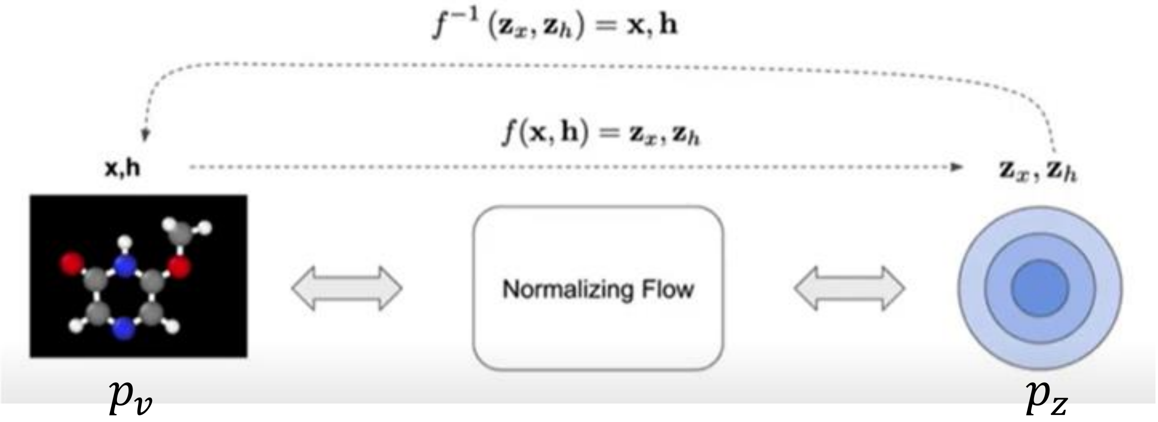
p_z : A simple invariant distribution (such as a Gaussian)

$$f_\theta(x,h) = z_x, z_h \quad g_\theta(z_x, z_h) = x, h$$

$$f_\theta = g_\theta^{-1}$$

$$p_V(\mathcal{V}) = p_V(\mathbf{x},\mathbf{h}) = p_Z(f_\theta(\mathbf{x},\mathbf{h}))|\det J_f| = p_Z(\mathbf{z}_x,\mathbf{z}_h)|\det J_f|, \tag{6}$$

E(n) Equivariant Normalizing Flows



$$p_V(\mathcal{V}) = p_V(\mathbf{x}, \mathbf{h}) = p_Z(f_\theta(\mathbf{x}, \mathbf{h}))|\det J_f| = p_Z(\mathbf{z}_x, \mathbf{z}_h)|\det J_f|, \tag{6}$$

- We require f to be invertible.
- We require f to be equivariant.

The solution to an ODE defined as:

$$\mathbf{z}_x, \mathbf{z}_h = f(\mathbf{x}, \mathbf{h}) = [\mathbf{x}(0), \mathbf{h}(0)] + \int_0^1 \phi(\mathbf{x}(t), \mathbf{h}(t))\mathrm{d}t. \tag{7}$$

$Where\ x(0) = x, h(0) = h;\ \ x(1) = z_x, h(1) = z_h$

E(n) Equivariant Normalizing Flows

The solution to an ODE defined as:

$$\mathbf{z}_x, \mathbf{z}_h = f(\mathbf{x}, \mathbf{h}) = [\mathbf{x}(0), \mathbf{h}(0)] + \int_0^1 \underbrace{\phi(\mathbf{x}(t), \mathbf{h}(t))}_{\text{EGNN, but not original EGNN}} \mathrm{d}t. \tag{7}$$

- The original EGNN from (Satorras et al., ICML 2021) is unstable when utilized in an ODE because the coordinate update from Equation 5 would easily explode

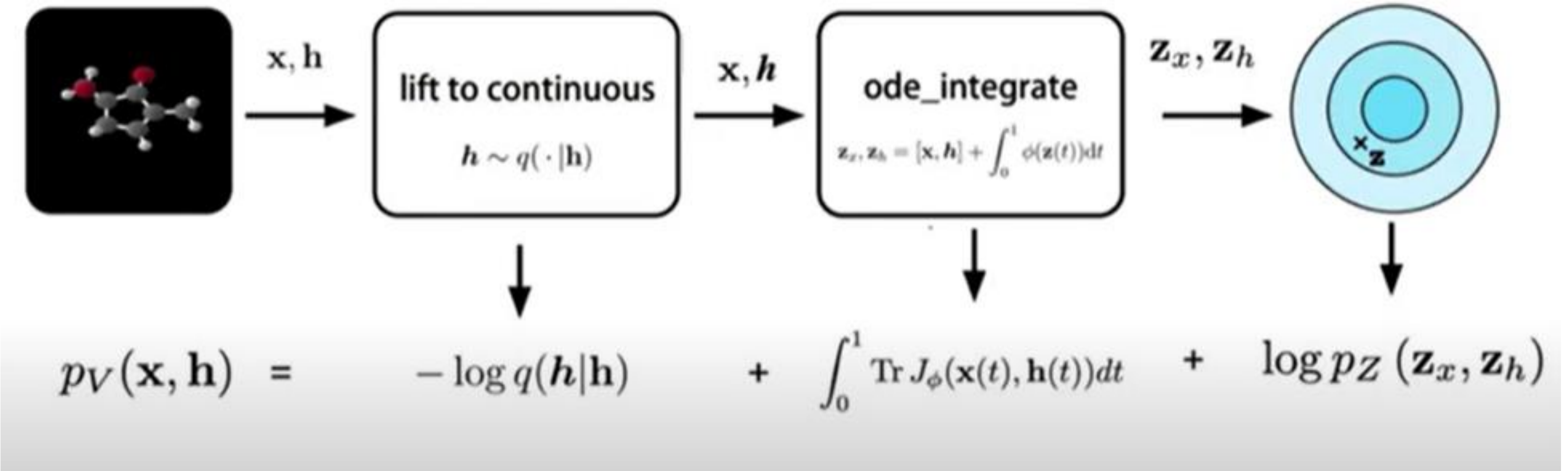
$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \left\|\mathbf{x}_i^l - \mathbf{x}_j^l\right\|^2\right) \quad \text{and} \quad \mathbf{m}_i = \sum_{j \neq i} e_{ij}\mathbf{m}_{ij}, \tag{4}$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \sum_{j \neq i}\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)\phi_x\left(\mathbf{m}_{ij}\right) \quad \text{and} \quad \mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right). \tag{5}$$

- An extension of the original EGNN from :

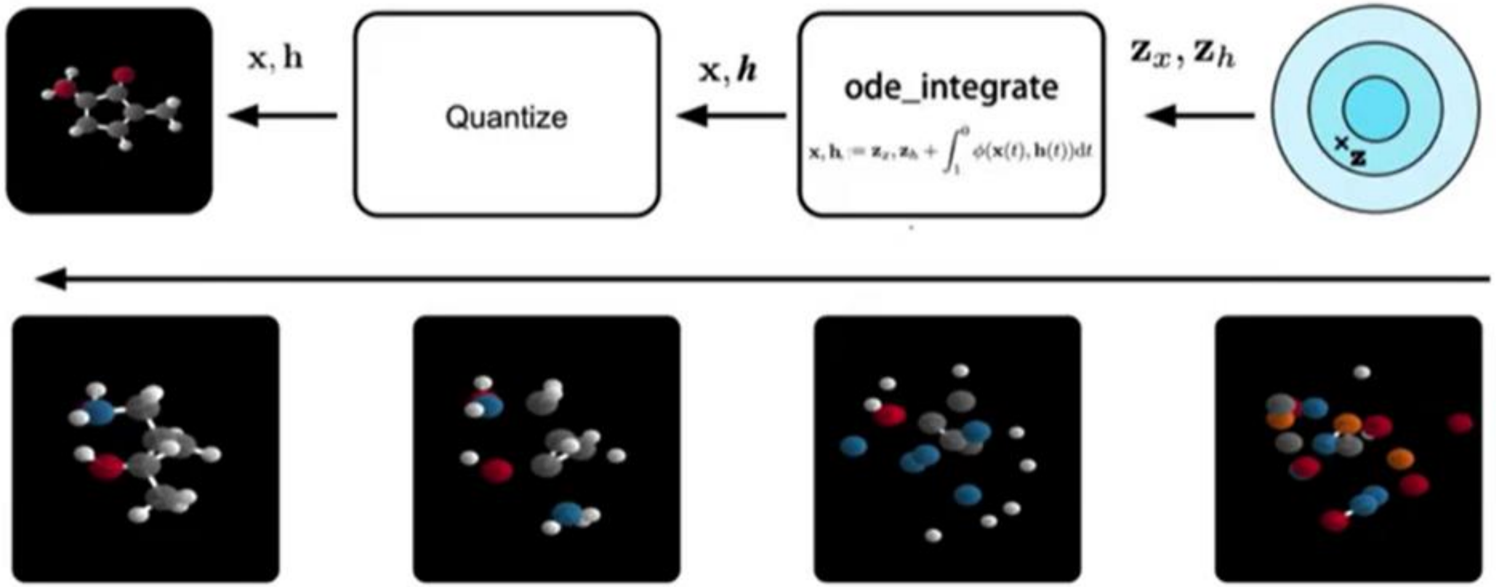
$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \sum_{j \neq i} \frac{\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\| + C}\phi_x\left(\mathbf{m}_{ij}\right) \tag{9}$$

E(n) Equivariant Normalizing Flows | Training process



- Sample a point (x, h) from the dataset
- The discrete h is lifted to continuous h
- The variables x, h are transformed by an ODE to z_x, z_h
- Get the loglikelihood $\log p_z(z_x, z_h)$ on a simple invariant distribution (such as a Gaussian)

E(n) Equivariant Normalizing Flows | Generation



- Sample a point (z_x, z_h) from the simple distribution
- The variables z_x, z_h are transformed by an ODE to x, h
- Quantize

Experiments

- We compare to the state-of-the art E(n) equivariant flows:
  - "Simple Dynamics"
  - "Kernel Dynamics" presented

- We also compare to non-equivariant variants of our method:
  - Graph Normalizing Flow (GNF)
  - GNF with attention (GNF-att)
  - GNF with attention and data augmentation (GNF-att-aug)

Experiments | DW4 && LJ13

DW4 and LJ13 have been synthetically generated by sampling from their respective energy functions using Markov Chain Monte Carlo (MCMC).

Table 1: Negative Log Likelihood comparison on the test partition over different methods on DW4 and LJ13 datasets for different amount of training samples averaged over 3 runs.

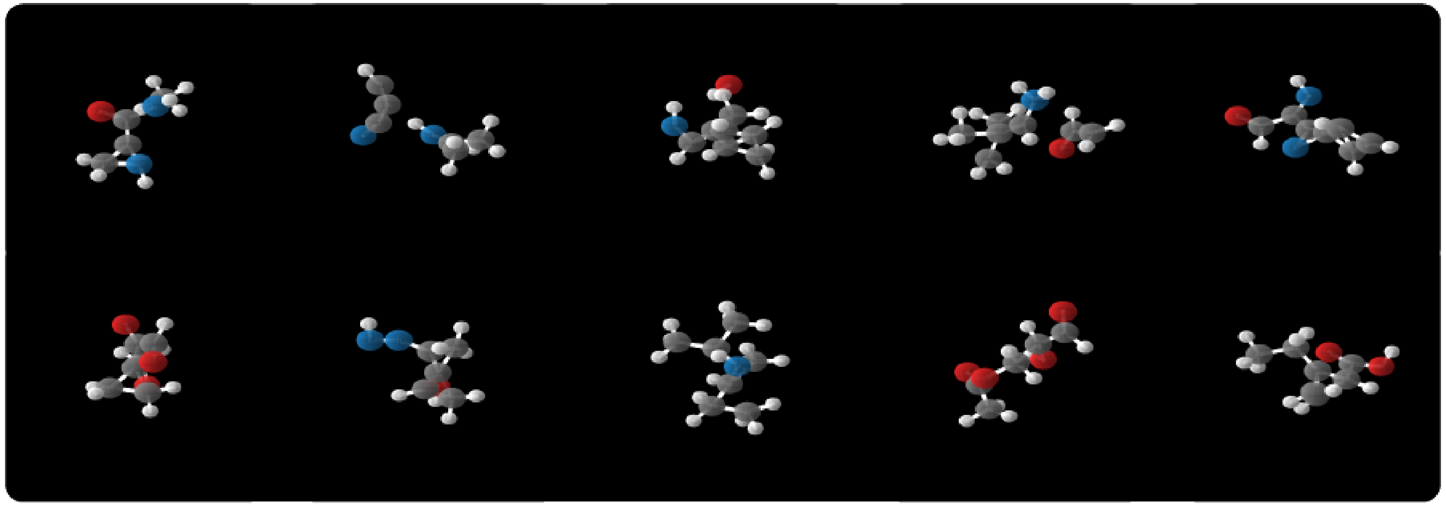
	DW4				**LJ13**			
# Samples	10^2	10^3	10^4	10^5	10	10^2	10^3	10^4
GNF	11.93	11.31	10.38	7.95	43.56	42.84	37.17	36.49
GNF-att	11.65	11.13	9.34	7.83	43.32	36.22	33.84	32.65
GNF-att-aug	8.81	8.31	7.90	7.61	41.09	31.50	30.74	30.93
Simple dynamics	9.58	9.51	9.53	9.47	33.67	33.10	32.79	32.99
Kernel dynamics	8.74	8.67	8.42	8.26	35.03	31.49	31.17	31.25
E-NF	**8.31**	**8.15**	**7.69**	**7.48**	**33.12**	**30.99**	**30.56**	**30.41**

Experiments | QM9

QM9 (Ramakrishnan et al., 2014) is a molecular dataset in machine learning as a chemical property prediction benchmark

Table 2: Neg. log-likelihood $-\log p_V(\mathbf{x}, \mathbf{h}, M)$, atom stability and mol stability for the QM9 dataset.

# Metrics	NLL	Atom stability	Mol stable
GNF-attention	-28.2	72%	0.3%
GNF-attention-augmentation	-29.3	75%	0.5%
E-NF (ours)	**-59.7**	**85%**	**4.9%**
Data	-	99%	95.2%

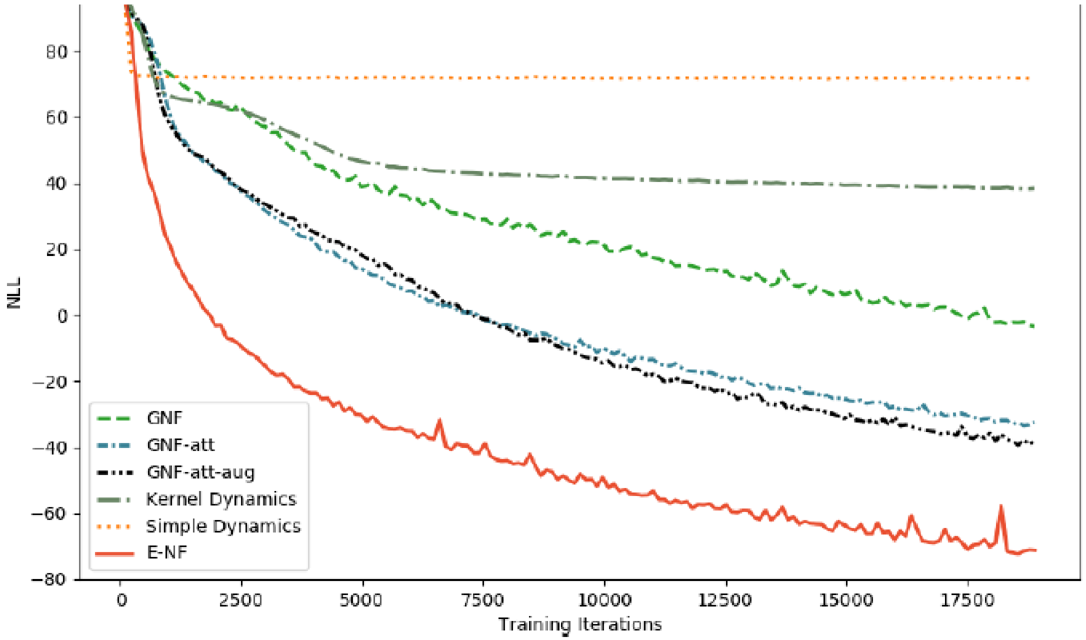


The top row contains random samples, the bottom row also contains samples but selected to be stable.

Experiments | QM9 Positional

We introduce **QM9 Positional** as a subset of QM9 that only considers positional information and does not encode node features

# Metrics	NLL	JS(rel. dist)
Simple dynamics	73.0	.086
Kernel dynamics	38.6	.029
GNF	-00.9	.011
GNF-att	-26.6	.007
GNF-att-aug	-33.5	.006
E-NF (ours)	**-70.2**	.006



Conclusion

- The ODE type of flow makes the training expensive and slow since the same forward operation has to be done multiple times.
- E(n) Equivariant Normalizing Flows (E-NFs) are continous-time normalizing flows that utilize an EGNN with improved stability as parametrization.
- This method may accelerate drug discovery by providing better drug candidate molecules.

Thanks!