Improving Equivariant Networks with Probabilistic Symmetry Breaking

Hannah Lawrence*, Vasco Portilheiro*, Yan Zhang, Sékou-Oumar Kaba

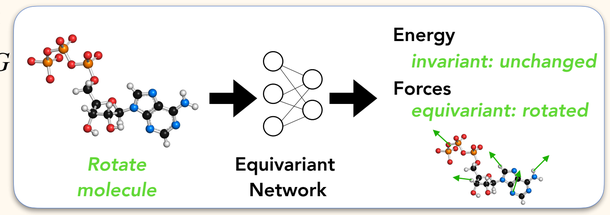
Background: Symmetries in ML

Equivariance:

$f : X \to Y, g \in G$

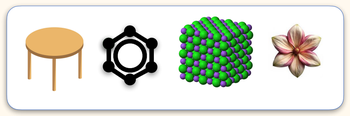
$f(gx) = gf(x)$

Benefits in generalization, sample complexity



Equivariant functions can't break symmetries…

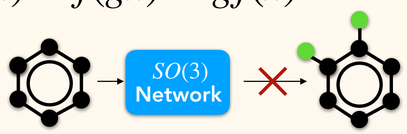
Many objects are themselves inherently symmetric:



Equivariance preserves data symmetries: for all group elements g,

$$x = gx \to f(x) = f(gx) = gf(x)$$

This is restrictive: an equivariant network can't turn a symmetric input into an asymmetric output



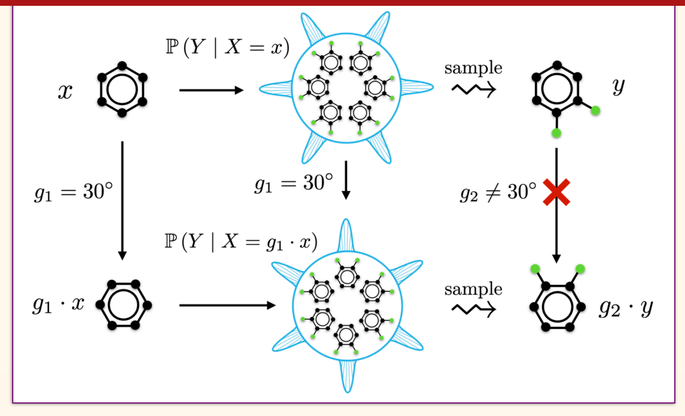
But: retaining the inductive bias of equivariance on asymmetric inputs is still desirable.

	Generalization benefits	Can break symmetries
Non-equivariant model	✗	✓
Equivariant model	✓	✗
Symmetry-breaking (goal)	✓	✓

How do we achieve this? First, observe that any of these outputs is equally good:

Suggests: learn a map to equivariant *distributions* $f : X \to \mathcal{P}(Y)$, such that individual samples from $f(x)$ can break the symmetry of x

Equivariant distributions can break symmetries



Equivariant networks **can't break symmetries**, which is **problematic for generative models**. We use symmetry-breaking positional encodings to ***minimally*** **break symmetry**.

Probabilistic Representation Theorem

***How can we learn* equivariant distributions?**

→ **requires a source of randomness ("noise outsourcing", e.g. [2])**

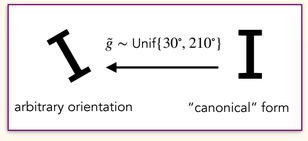
Theorem: $Y \,|\, X$ is equivariant iff, for some $f : X \times G \times (0,1) \to Y$ jointly equivariant in X and g (i.e. $f(hx, hg, \epsilon) = hf(x, g, \epsilon)$), $\epsilon \sim \text{Unif}(0,1)$, and $\tilde{g} \,|\, X$ as defined below,

$$Y \overset{a.s.}{=} f(X, \tilde{g}, \epsilon)$$

Asymmetric output · *Symmetric input* · *"Equivariant randomness" to break symmetries* · *"Arbitrary randomness"*

$\tilde{g}$ should have as little randomness as possible to distinguish among the symmetrically equivalent outputs, e.g.

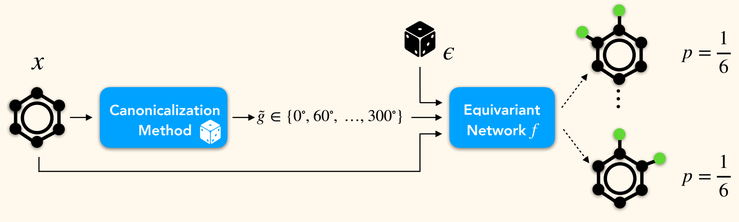
Canonicalization conveniently provides this randomness



Generalization to "noise injection": can let $\tilde{g}$ more generally be a random variable with no self-symmetries and $\tilde{g} \,|\, X \sim h\tilde{g} \,|\, hX$

→ Sometimes easier to compute, e.g. to avoid solving graph automorphism! Also, applies to [3]

Symmetry-Breaking Positional Encodings



To make an equivariant network f break symmetries, we use canonicalization to sample a random $\tilde{g}$ and add it as input

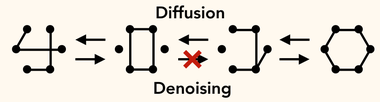
Algorithm 1 SymPE: Symmetry-Breaking Positional Encodings

Inputs: input $x \in \mathcal{X}$
Learnable parameters: learned vector $v \in \mathcal{V}$, equivariant neural network parameters θ, canonicalization parameters ϕ
Sample $\tilde{g} \sim h_\phi(x)$ ▷ Sample group element for canonicalization
$\tilde{v} \leftarrow \tilde{g}v$ ▷ Apply group element to learned vector
Return $f_\theta(x \oplus \tilde{v})$ ▷ Forward pass with positional encoding

Experiments

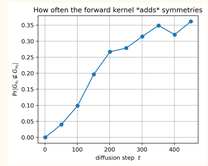
Graph Generation with Diffusion Models

Problem: Noising process is likely to introduce symmetries that cannot be denoised



Experiment: DiGress discrete diffusion with graph transformer, using SymPE with sorting canonicalization to break symmetries

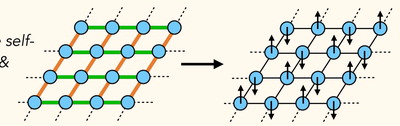
Method	NLL
DiGress	129.7
DiGress + noise	126.5
DiGress + SymPE	30.3

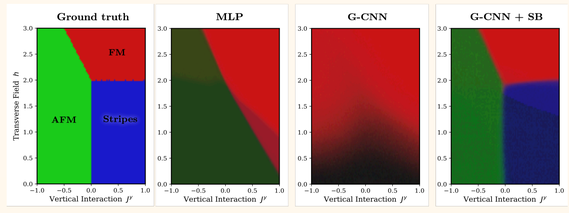


Ising Model

Task: Given vertical and horizontal interaction strengths (J_x, J_y), output a configuration (up or down spin for each node in the grid graph) of minimal energy

Interaction strengths have self-symmetry: reflections & translations · *Energy function has symmetry: rotations, translations, and reflections*

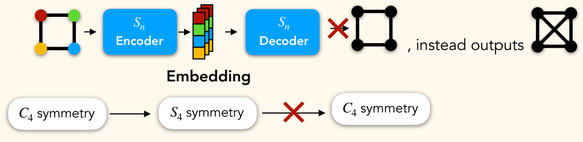




Phase of predicted ground states. Green: antiferromagnetic, Red: ferromagnetic, Blue: Stripes, Black: No order.

Graph Autoencoder

Problem: An S_n-equivariant encoder with per-node embeddings can spuriously introduce extra symmetry, which an S_n-equivariant decoder cannot break.



Solution: Break symmetries via sorting a learned scalar embedding per-node, where sorting breaks ties

Experiment: Erdös-Renyi random graphs from [4] with GNN auto encoder architecture

Method	% Correct Edges
No symmetry-breaking	3.9
Noise (randomly initialized node features)	2.3
Randomly permuting instead of sorting	1.3
Passing in Laplacian encodings directly (no sorting)	0.98
Our method: sorting learned embeddings	0.77

References

1.S.-O. Kaba and S. Ravanbakhsh. Symmetry breaking and equivariant neural networks, 2023.
2. B. Bloem-Reddy and Y. W. Teh. Probabilistic symmetries and invariant neural networks, 2020
3. Y. Xie and T. Smidt. Equivariant symmetry breaking sets, 2024.
4. V. G. Satorras, E. Hoogeboom, and M. Welling. E(n) equivariant graph neural networks, 2021
5. C. Vignac et al. Digress: Discrete denoising diffusion for graph generation, 2022.