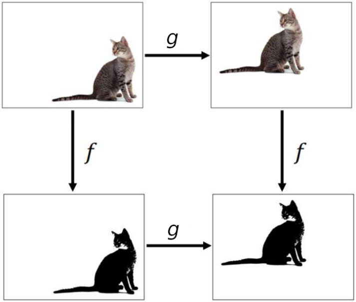
Stochastic Neural Network Symmetrisation in Markov Categories

Rob Cornish

Department of Statistics, University of Oxford

September 23, 2025

Motivation: symmetry



https://www.doc.ic.ac.uk/~bkainz/teaching/DL/notes/equivariance.pdf

Formulation

A neural network $f : X \to Y$ is equivariant with respect to the actions of a group G if

$$f(g \cdot x) = g \cdot f(x)$$

for all $x \in X$ and $g \in G$

Formulation

A neural network $f : X \to Y$ is equivariant with respect to the actions of a group G if

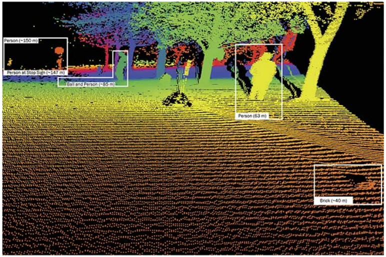
$$f(g \cdot x) = g \cdot f(x)$$

for all $x \in X$ and $g \in G$

In this example:

- X is set of images
- Y is set of binarisations
- G is the group of translations

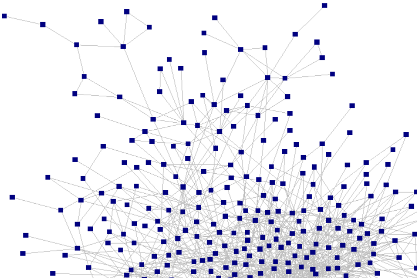
Many other examples



photonics.com

alphafold.ebi.ac.uk

orgnet.com



Key question

How can we parameterise an equivariant neural network?

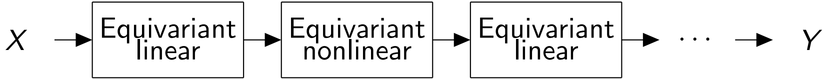
Key question

How can we parameterise an equivariant neural network?

Two key approaches: intrinsic equivariance and symmetrisation

Intrinsic equivariance

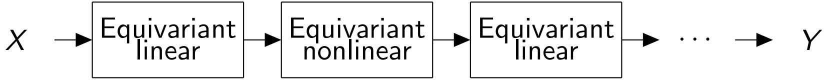
Overall model $f : X \to Y$ has form



where the individual layers are all equivariant via e.g. weight tying

Intrinsic equivariance

Overall model $f : X \to Y$ has form



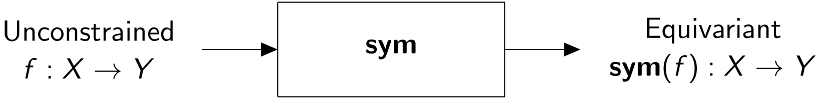
where the individual layers are all equivariant via e.g. weight tying

A natural idea, but:

- Requires hand engineering for each case
- Nonlinear layers are often ad hoc
- Can be brittle at scale (e.g. AlphaFold 2 vs. 3)

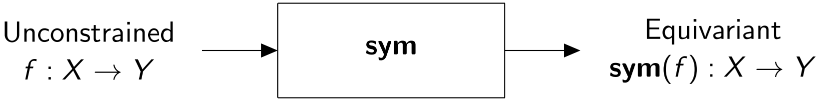
Symmetrisation

Recent interest instead in symmetrisation:



Symmetrisation

Recent interest instead in symmetrisation:



Here f is completely general and opaque

Symmetrisation: example

Early example is Janossy pooling [Murphy et al., 2019]: given

$$f : X^n \to \mathbb{R}^d,$$

the following function $X^n \to \mathbb{R}^n$ is always permutation invariant:

$$\frac{1}{n!} \sum_{\sigma \in S_n} f(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$$

Symmetrisation: other examples

Other recent examples, given $f : X \to \mathbb{R}^d$ and a group G

$\frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} g \cdot f(g^{-1} \cdot x)$ [Puny et al., 2022]

$h(x) \cdot f(h(x)^{-1} \cdot x)$ [Kaba et al., 2023]

$\mathbb{E}_{\boldsymbol{G} \sim p(g|x)}[\boldsymbol{G} \cdot f(\boldsymbol{G}^{-1} \cdot x)]$ [Kim et al., 2023]

Symmetrisation: other examples

Other recent examples, given $f : X \to \mathbb{R}^d$ and a group G

$\frac{1}{|\mathcal{F}(x)|} \sum_{g \in \mathcal{F}(x)} g \cdot f(g^{-1} \cdot x)$ [Puny et al., 2022]

$h(x) \cdot f(h(x)^{-1} \cdot x)$ [Kaba et al., 2023]

$\mathbb{E}_{\boldsymbol{G} \sim p(g|x)}[\boldsymbol{G} \cdot f(\boldsymbol{G}^{-1} \cdot x)]$ [Kim et al., 2023]

Under some conditions, each is equivariant in $x \in X$, even if f is arbitrarily complex

Some research questions

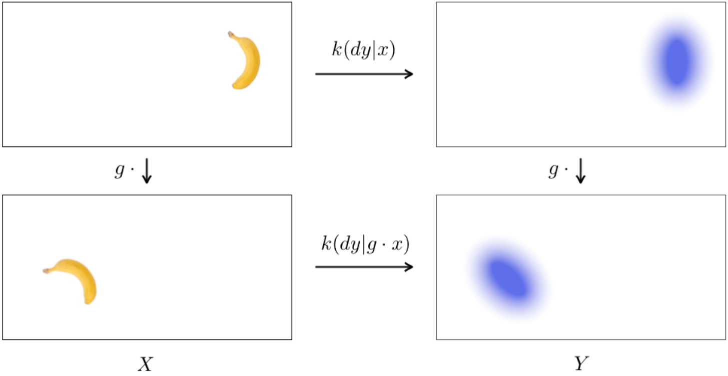
Where do these methods "come from"? Are they the only possibilities?

Some research questions

Where do these methods “come from”? Are they the only possibilities?

What about stochastic models?

Stochastic equivariance: illustration



Contribution

Methodological contribution

Stochastic Neural Network Symmetrisation in Markov Categories

Rob Cornish
Department of Statistics, University of Oxford

Contribution

A general theory of symmetrisation procedures that extends to stochastic models (plus various other methodological extensions)

Theoretical contribution

Underlying theory of [Cornish, 2024] is developed in terms of Markov categories

Theoretical contribution

Underlying theory of [Cornish, 2024] is developed in terms of Markov categories

Implication

Markov categories can produce novel methodology for AI (not just retrospective simplifications)

Theoretical contribution

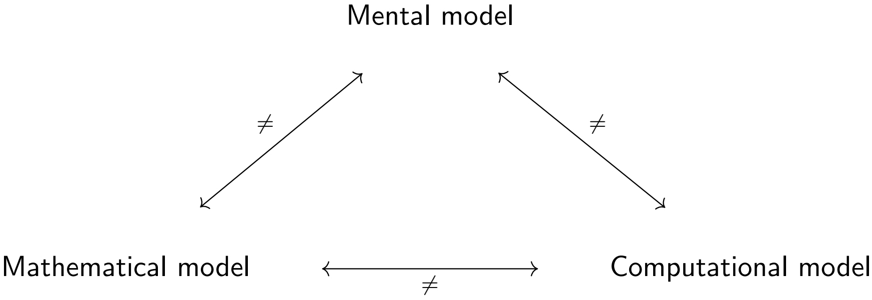
Underlying theory of [Cornish, 2024] is developed in terms of Markov categories

Implication

Markov categories can produce novel methodology for AI (not just retrospective simplifications)

But why care in the first place?

Digression: three models



Probabilistic reasoning

For probabilistic settings, a major reason for this is measure theory

Probabilistic reasoning

For probabilistic settings, a major reason for this is measure theory

In practice, we often prefer semi-formal "density" notation, e.g.

$$p(x, y) = p(x)\, p(y|x)$$

Probabilistic reasoning

For probabilistic settings, a major reason for this is measure theory

In practice, we often prefer semi-formal "density" notation, e.g.

$$p(x, y) = p(x)\, p(y|x)$$

Works well in many cases, but have to write things like

$$x \sim p_\theta(x|z \sim q_\phi(z|x, y), y)$$

which can make things actually more complex

Example: stochastic equivariance in densities

A density $p(y|x)$ is equivariant if (provided $g\cdot$ has unit Jacobian)

$$p(y|x) = p(g \cdot y | g \cdot x)$$

Example: stochastic equivariance in densities

A density $p(y|x)$ is equivariant if (provided $g\cdot$ has unit Jacobian)

$$p(y|x) = p(g \cdot y | g \cdot x)$$

Hard to see the input/output interpretation of equivariance here

The Markov categorical approach

Markov categories abstract away painful technical details, but maintains rigour

The Markov categorical approach

Markov categories abstract away painful technical details, but maintains rigour

Theory becomes very conceptual and diagrammatic, and closer to mental and computational models (e.g. DisCoPy)

The Markov categorical approach

Markov categories abstract away painful technical details, but maintains rigour

Theory becomes very conceptual and diagrammatic, and closer to mental and computational models (e.g. DisCoPy)

Empirically, this was actually how this work came about!

Symmetrisation procedures in Markov categories

Markov kernels

The key example of a Markov category is **Stoch**:

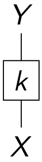
- Objects X and Y are measurable spaces
- Morphisms $k : X \to Y$ are Markov kernels $k(dy|x)$

Markov kernels

The key example of a Markov category is **Stoch**:

- Objects X and Y are measurable spaces
- Morphisms $k : X \to Y$ are Markov kernels $k(dy|x)$

Think of kernels as conditional distributions or stochastic maps

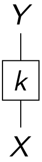


Markov kernels

The key example of a Markov category is **Stoch**:

- Objects X and Y are measurable spaces
- Morphisms $k : X \to Y$ are Markov kernels $k(dy|x)$

Think of kernels as conditional distributions or stochastic maps



Can formalise as functions $k : \Sigma_Y \times X \to [0, 1]$ satisfying some conditions

Markov categories

Definition ([Fritz, 2020], [Cho and Jacobs, 2019])

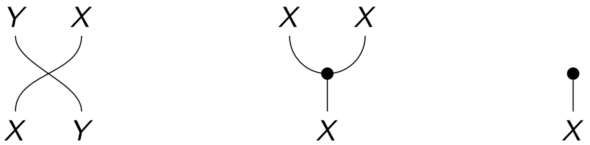
A Markov category is a semicartesian symmetric monoidal category $(\mathbf{C}, \otimes, I)$ in which every object X is equipped with a commutative comonoid structure $(\mathbf{copy}_X, \mathbf{del}_X)$ that is suitably compatible with $\otimes$.

Markov categories

Definition ([Fritz, 2020], [Cho and Jacobs, 2019])

A Markov category is a semicartesian symmetric monoidal category $(\mathbf{C}, \otimes, I)$ in which every object X is equipped with a commutative comonoid structure $(\mathbf{copy}_X, \mathbf{del}_X)$ that is suitably compatible with $\otimes$.

Essentially, we can compose sequentially and in parallel, and can swap, copy, and discard information:



Examples of Markov categories

Many examples of Markov categories including:

Category	Objects	Morphisms
Stoch	Measurable spaces	Markov kernels
BorelStoch	Standard Borel spaces	Markov kernels
TopStoch	Topological spaces	Continuous Markov kernels
⋮		

Examples of Markov categories

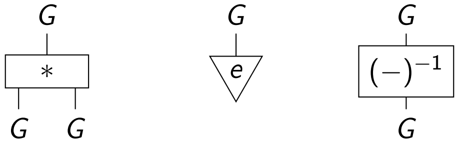
Many examples of Markov categories including:

Category	Objects	Morphisms
Stoch	Measurable spaces	Markov kernels
BorelStoch	Standard Borel spaces	Markov kernels
TopStoch	Topological spaces	Continuous Markov kernels
⋮		
Set	Sets	Functions
Meas	Measurable spaces	Measurable functions
Top	Topological spaces	Continuous functions

Theory is now “write once, run anywhere”

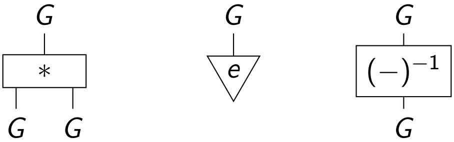
Groups and actions

A group in a Markov category **C** is an object G equipped with

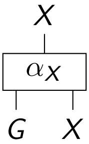


Groups and actions

A group in a Markov category **C** is an object G equipped with

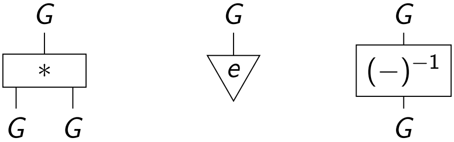


An action of a group G is a morphism

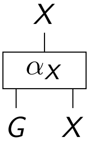


Groups and actions

A group in a Markov category **C** is an object G equipped with



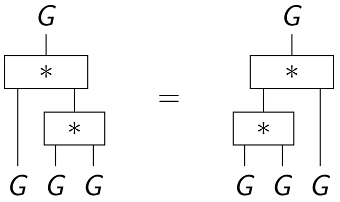
An action of a group G is a morphism



Both satisfy the usual axioms (expressed in diagrams)

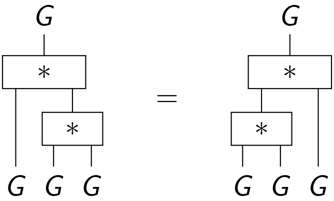
Example: associativity

Multiplication must satisfy:



Example: associativity

Multiplication must satisfy:

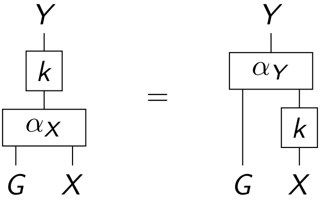


In **Set**, this just recovers associativity: for all $g, g', g'' \in G$ we have

$$g(g'g'') = (gg')g''$$

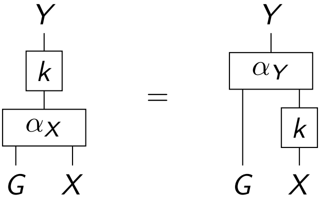
Equivariance

A morphism $k : X \to Y$ is equivariant with respect to α_X and α_Y if



Equivariance

A morphism $k : X \to Y$ is equivariant with respect to α_X and α_Y if

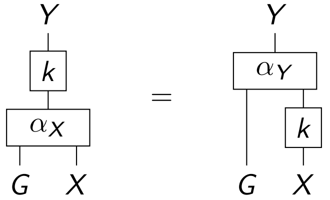


When the morphisms of **C** are functions, this gives the usual

$$k(g \cdot x) = g \cdot k(x)$$

Equivariance

A morphism $k : X \to Y$ is equivariant with respect to α_X and α_Y if



When the morphisms of **C** are functions, this gives the usual

$$k(g \cdot x) = g \cdot k(x)$$

For Markov kernels, this gives stochastic equivariance:

$$k(dy|g \cdot x) = g \cdot k(dy|x)$$

Markov category of equivariant maps

Theorem

Given a group G in a Markov category $\mathbf{C}$, always obtain a Markov category $\mathbf{C}^G$ where:

Markov category of equivariant maps

Theorem

Given a group G in a Markov category $\mathbf{C}$, always obtain a Markov category $\mathbf{C}^G$ where:

- *Objects are pairs (X, α_X), where α_X is an action of G on X*

Markov category of equivariant maps

Theorem

Given a group G in a Markov category $\mathbf{C}$, always obtain a Markov category $\mathbf{C}^G$ where:

- *Objects are pairs (X, α_X), where α_X is an action of G on X*
- *Morphisms $(X, \alpha_X) \to (Y, \alpha_Y)$ are equivariant w.r.t. α_X and α_Y*

Markov category of equivariant maps

Theorem

Given a group G in a Markov category $\mathbf{C}$, always obtain a Markov category $\mathbf{C}^G$ where:

- *Objects are pairs (X, α_X), where α_X is an action of G on X*
- *Morphisms $(X, \alpha_X) \to (Y, \alpha_Y)$ are equivariant w.r.t. α_X and α_Y*
- *Other components ($\otimes$, copy maps, etc.) are inherited from $\mathbf{C}$*

Symmetrisation procedures

Definition (for today)

A symmetrisation procedure is a function **sym** of the following form

$$\underbrace{\mathbf{C}(X,Y)}_{\text{Morphisms } X \to Y \text{ in } \mathbf{C}} \xrightarrow{\textbf{sym}} \mathbf{C}^G((X,\alpha_X),(Y,\alpha_Y))$$

Symmetrisation procedures

Definition (for today)

A symmetrisation procedure is a function **sym** of the following form

$$\underbrace{\mathbf{C}(X, Y)}_{\text{Morphisms } X \to Y \text{ in } \mathbf{C}} \xrightarrow{\textbf{sym}} \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y))$$

Despite generality, can characterise all such functions of this form

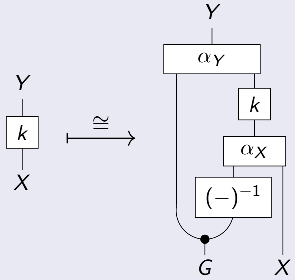
Key result

Theorem

There is always a bijection

$$\mathbf{C}(X, Y) \xrightarrow{\cong} \mathbf{C}^G((G,*) \otimes (X, \alpha_X), (Y, \alpha_Y))$$

defined as follows:



Categorical explanation

Arises from an adjunction of the form

$$\mathbf{C} \underset{U}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{C}^G$$

where $U(X, \alpha_X) := X$, which gives

$$\begin{aligned} \mathbf{C}(X, Y) &= \mathbf{C}(U(X, \alpha_X), U(Y, \alpha_Y)) \\ &\cong \mathbf{C}^G(FU(X, \alpha_X), (Y, \alpha_Y)) \\ &\cong \mathbf{C}^G((G, *) \otimes (X, \alpha_X), (Y, \alpha_Y)) \end{aligned}$$

A general strategy for symmetrisation

Corollary

Every symmetrisation procedure $\mathbf{C}(X, Y) \xrightarrow{\mathbf{sym}} \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y))$ *can be expressed as a composition*

$$\mathbf{C}(X, Y) \xrightarrow{\cong} \mathbf{C}^G((G, *) \otimes (X, \alpha_X), (Y, \alpha_Y)) \longrightarrow \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y)),$$

and vice versa, for some choice of function in the second step.

A general strategy for symmetrisation

Corollary

Every symmetrisation procedure $\mathbf{C}(X, Y) \xrightarrow{\textbf{sym}} \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y))$ *can be expressed as a composition*

$$\mathbf{C}(X, Y) \xrightarrow{\cong} \mathbf{C}^G((G, *) \otimes (X, \alpha_X), (Y, \alpha_Y)) \longrightarrow \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y)),$$

and vice versa, for some choice of function in the second step.

Only (natural) choice for second step is precomposition:

$$(G, *) \otimes (X, \alpha_X) \xrightarrow{k} (Y, \alpha_Y)$$

A general strategy for symmetrisation

Corollary

Every symmetrisation procedure $\mathbf{C}(X, Y) \xrightarrow{\textbf{sym}} \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y))$ *can be expressed as a composition*

$$\mathbf{C}(X, Y) \xrightarrow{\cong} \mathbf{C}^G((G, *) \otimes (X, \alpha_X), (Y, \alpha_Y)) \longrightarrow \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y)),$$

and vice versa, for some choice of function in the second step.

Only (natural) choice for second step is precomposition:

$$(X, \alpha_X) \xrightarrow{\Gamma} (G, *) \otimes (X, \alpha_X) \xrightarrow{k} (Y, \alpha_Y)$$

i.e. $k \mapsto k \circ \Gamma$

Precomposition morphism

Natural to require that if k is already G-equivariant, then

$$\mathbf{sym}(k) = k$$

i.e. procedure is stable on equivariant inputs

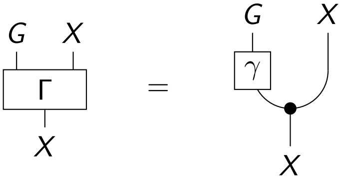
Precomposition morphism

Natural to require that if k is already G-equivariant, then

$$\mathbf{sym}(k) = k$$

i.e. procedure is stable on equivariant inputs

Can show: holds iff precomposition morphism has the form



where $\gamma : (X, \alpha_X) \to (G, *)$ in $\mathbf{C}^G$

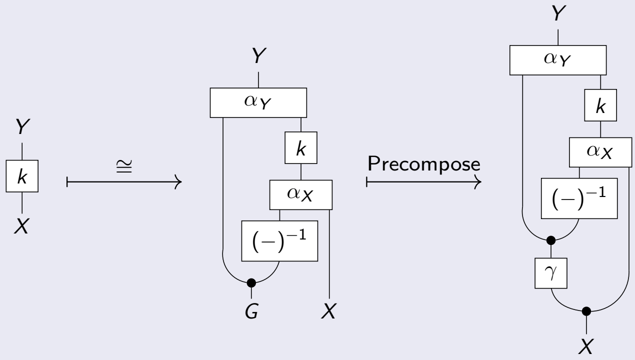
End-to-end procedure

Algorithm

Given a suitable γ, overall procedure now has form

$$\mathbf{C}(X, Y) \xrightarrow{\cong} \mathbf{C}^G((G,*) \otimes (X, \alpha_X), (Y, \alpha_Y)) \longrightarrow \mathbf{C}^G((X, \alpha_X), (Y, \alpha_Y))$$

where these steps are computed as follows:



Instantiation in **Set**

Corollary

Suppose G is a group acting on X and Y. If $k : X \to Y$ is any function, and $\gamma : X \to G$ is equivariant (where G acts on itself by left multiplication), then the following defines an equivariant function $X \to Y$ given $x \in X$:

$$\gamma(x) \cdot k(\gamma(x)^{-1} \cdot x)$$

Instantiation in **Set**

Corollary

Suppose G is a group acting on X and Y. If $k : X \to Y$ is any function, and $\gamma : X \to G$ is equivariant (where G acts on itself by left multiplication), then the following defines an equivariant function $X \to Y$ given $x \in X$:

$$\gamma(x) \cdot k(\gamma(x)^{-1} \cdot x)$$

Exactly recovers canonicalisation [Kaba et al., 2023]

Instantiation in **Stoch**

Also obtain a novel procedure for stochastic symmetrisation

Instantiation in **Stoch**

Also obtain a novel procedure for stochastic symmetrisation

Corollary

Suppose G is a measurable group acting measurably on X and Y. If $k : X \to Y$ is any Markov kernel, and $\gamma : X \to G$ is stochastically equivariant (where G acts on itself by left multiplication), then the following sampling process given $x \in X$ defines a stochastically equivariant Markov kernel $X \to Y$:

$$\boldsymbol{G} \sim \gamma(dg|x) \qquad \boldsymbol{Y} \sim k(dy|\boldsymbol{G}^{-1} \cdot x) \qquad \textit{return } \boldsymbol{G} \cdot \boldsymbol{Y}$$

Instantiation in **Stoch**

Also obtain a novel procedure for stochastic symmetrisation

Corollary

Suppose G is a measurable group acting measurably on X and Y. If $k : X \to Y$ is any Markov kernel, and $\gamma : X \to G$ is stochastically equivariant (where G acts on itself by left multiplication), then the following sampling process given $x \in X$ defines a stochastically equivariant Markov kernel $X \to Y$:

$$\boldsymbol{G} \sim \gamma(dg|x) \qquad \boldsymbol{Y} \sim k(dy|\boldsymbol{G}^{-1} \cdot x) \qquad \textit{return } \boldsymbol{G} \cdot \boldsymbol{Y}$$

Note: technically should define this kernel as a function $\Sigma_Y \times X \to [0, 1]$ satisfying a measurability condition...

Extensions

The paper contains various extensions:

- Deterministic symmetrisation via averaging
- Symmetrisation along a homomorphism $\varphi : H \to G$
- Compositional usage
- Recursive usage to obtain γ

Extensions

The paper contains various extensions:

- Deterministic symmetrisation via averaging
- Symmetrisation along a homomorphism $\varphi : H \to G$
- Compositional usage
- Recursive usage to obtain γ

Also many examples:

- Compact groups
- Translation groups
- Direct and semidirect products
- Even $\mathrm{GL}(d, \mathbb{R})$

Extensions

The paper contains various extensions:

- Deterministic symmetrisation via averaging
- Symmetrisation along a homomorphism $\varphi : H \to G$
- Compositional usage
- Recursive usage to obtain γ

Also many examples:

- Compact groups
- Translation groups
- Direct and semidirect products
- Even $\mathrm{GL}(d, \mathbb{R})$

Markov categories allow describing all this in a uniform and coherent way

Numerical results

Follow-up work

SymDiff: Equivariant Diffusion via Stochastic Symmetrisation

Leo Zhang, Kianoosh Ashouritaklimi, Yee Whye Teh, Rob Cornish
Department of Statistics, University of Oxford

Overview

Recall that denoising diffusion models consist of forward and backwards processes defined as

$$q(\mathbf{z}_{0:T}) = q(\mathbf{z}_0) \prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{z}_{t-1}) \qquad p_\theta(\mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$$

Overview

Recall that denoising diffusion models consist of forward and backwards processes defined as

$$q(\mathbf{z}_{0:T}) = q(\mathbf{z}_0) \prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{z}_{t-1}) \qquad p_\theta(\mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$$

The idea is:

- $q(\mathbf{z}_0)$ is the data distribution
- $q(\mathbf{z}_T) \approx p(\mathbf{z}_T)$ is Gaussian
- Try to learn $p_\theta(\mathbf{z}_0) \approx q(\mathbf{z}_0)$

Overview

Recall that denoising diffusion models consist of forward and backwards processes defined as

$$q(\mathbf{z}_{0:T}) = q(\mathbf{z}_0) \prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{z}_{t-1}) \qquad p_\theta(\mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$$

The idea is:

- $q(\mathbf{z}_0)$ is the data distribution
- $q(\mathbf{z}_T) \approx p(\mathbf{z}_T)$ is Gaussian
- Try to learn $p_\theta(\mathbf{z}_0) \approx q(\mathbf{z}_0)$

Often want $p_\theta(\mathbf{z}_{t-1} | \mathbf{z}_t)$ to be equivariant (e.g. molecular data)

Strategy for equivariant diffusion

Previous work has enforced stochastic equivariance by setting

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) := \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t), \sigma_t^2 I)$$

where μ_θ is intrinsically equivariant (e.g. a graph neural network)

Strategy for equivariant diffusion

Previous work has enforced stochastic equivariance by setting

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) := \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t), \sigma_t^2 I)$$

where μ_θ is intrinsically equivariant (e.g. a graph neural network)

We instead take

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) := \mathbf{sym}_{\gamma_\theta}(k_\theta)(\mathbf{z}_{t-1}|\mathbf{z}_t)$$

where k_θ and γ_θ may leverage arbitrary neural networks

SymDiff training for $E(3)$-equivariance

Algorithm 1 SYMDIFF training step

1: Sample $\mathbf{z}_0 \sim p_{\text{data}}(\mathbf{z}_0)$, $t \sim \text{Unif}(\{1, \ldots, T\})$ and $\epsilon \sim \mathcal{N}_{\mathcal{U}}(0, \mathbf{I})$
2: $\mathbf{z}_t \leftarrow \alpha_t \mathbf{z}_0 + \sigma_t \epsilon$
3: Sample R_0 from the Haar measure on $\text{O}(3)$ and $\eta \sim \nu(d\eta)$
4: $R \leftarrow R_0 \cdot f_\theta(R_0^T \cdot \mathbf{z}_t, \eta)$
5: Take gradient descent step with $\nabla_\theta \frac{1}{2} w(t) \left\| \epsilon - R \cdot \epsilon_\theta(R^T \cdot \mathbf{z}_t) \right\|^2$

SymDiff training for $E(3)$-equivariance

Algorithm 1 SymDiff training step

1: Sample $\mathbf{z}_0 \sim p_{\text{data}}(\mathbf{z}_0)$, $t \sim \text{Unif}(\{1, \ldots, T\})$ and $\epsilon \sim \mathcal{N}_{\mathcal{U}}(0, \mathbf{I})$
2: $\mathbf{z}_t \leftarrow \alpha_t \mathbf{z}_0 + \sigma_t \epsilon$
3: Sample R_0 from the Haar measure on $\text{O}(3)$ and $\eta \sim \nu(d\eta)$
4: $R \leftarrow R_0 \cdot f_\theta(R_0^T \cdot \mathbf{z}_t, \eta)$
5: Take gradient descent step with $\nabla_\theta \frac{1}{2} w(t) \left\| \epsilon - R \cdot \epsilon_\theta(R^T \cdot \mathbf{z}_t) \right\|^2$

Resembles a learned data augmentation that is deployed at sampling time

Results

We obtained better performance compared with an intrinsic baseline (EDM [Hoogeboom et al., 2022]), and on par or better results compared with more sophisticated molecular models

Results

We obtained better performance compared with an intrinsic baseline (EDM [Hoogeboom et al., 2022]), and on par or better results compared with more sophisticated molecular models

Table 1: Test NLL, atom stability, molecular stability, validity and uniqueness on QM9 for 10,000 samples and 3 evaluation runs. We omit the results for NLL where not available.

Method	NLL ↓	Atm. stability (%) ↑	Mol. stability (%) ↑	Val. (%) ↑	Uniq. (%) ↑
GeoLDM	–	98.90 $\pm$0.10	89.40 $\pm$0.50	93.80 $\pm$0.40	92.70 $\pm$0.50
MUDiff	**-135.50**$\pm$2.10	98.80$\pm$0.20	**89.90**$\pm$1.10	95.30$\pm$1.50	**99.10**$\pm$0.50
END	–	98.90 $\pm$0.00	89.10 $\pm$0.10	94.80 $\pm$0.10	92.60 $\pm$0.20
EDM	-110.70$\pm$1.50	98.70$\pm$0.10	82.00 $\pm$0.40	91.90$\pm$0.50	90.70$\pm$0.60
SymDiff*	-133.79$\pm$1.33	**98.92**$\pm$0.03	89.65$\pm$0.10	**96.36**$\pm$0.27	97.66$\pm$0.22
SymDiff	-129.35 $\pm$1.07	98.74$\pm$0.03	87.49 $\pm$0.23	95.75 $\pm$0.10	97.89 $\pm$0.26
SymDiff-H	-126.53$\pm$0.90	98.57$\pm$0.07	85.51$\pm$0.18	95.22$\pm$0.18	97.98$\pm$0.09
DiT-Aug	-126.81$\pm$1.69	98.64$\pm$0.03	85.85$\pm$0.24	95.10$\pm$0.17	97.98$\pm$0.08
DiT	-127.78$\pm$2.49	98.23$\pm$0.04	81.03$\pm$0.25	94.71$\pm$0.31	97.98$\pm$0.12
Data		99.00	95.20	97.8	100

Thank you!

References I

Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BJluy2RcFm`.

Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=zIUyj55nXR`.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15546–15566. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/kaba23a.html`.

References II

Jinwoo Kim, Dat Nguyen, Ayhan Suleymanzade, Hyeokjun An, and Seunghoon Hong. Learning probabilistic symmetrization for architecture agnostic equivariance. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 18582–18612. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3b5c7c9c5c7bd77eb73d0baec7a07165-Paper-Conference.pdf.

Rob Cornish. Stochastic neural network symmetrisation in markov categories, 2024. URL https://arxiv.org/abs/2406.11814.

Tobias Fritz. A synthetic approach to markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, August 2020. ISSN 0001-8708. doi: 10.1016/j.aim.2020.107239. URL http://dx.doi.org/10.1016/j.aim.2020.107239.

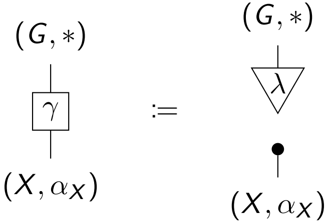
Kenta Cho and Bart Jacobs. Disintegration and bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29(7):938–971, March 2019. ISSN 1469-8072. doi: 10.1017/s0960129518000488. URL http://dx.doi.org/10.1017/S0960129518000488.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8867–8887. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hoogeboom22a.html.

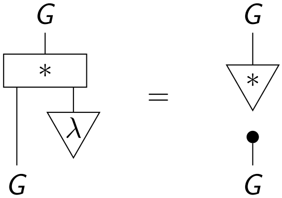
Appendix

Example γ

When G is compact, can choose $\gamma : (X, \alpha_X) \to (G, *)$ as



where here $\lambda : (I, \epsilon) \to (G, *)$ satisfies



Determinism via averaging

Proposition

Suppose $Y = \mathbb{R}^d$, and denote

$$\mathbf{ave}(k)(x) := \int y\, k(dy|x)$$

If G acts linearly on Y, then this corresponds to a function

$$\mathbf{Stoch}^G((X,\alpha_X),(Y,\alpha_Y)) \xrightarrow{\mathbf{ave}} \mathbf{Stoch}^G_{\mathrm{det}}((X,\alpha_X),(Y,\alpha_Y)).$$

Deterministic symmetrisation via averaging

Can combine averaging with stochastic symmetrisation:

$$\textbf{Stoch}(X, Y) \xrightarrow{\textsf{sym}_\gamma} \textbf{Stoch}^G((X, \alpha_X), (Y, \alpha_Y)) \\ \xrightarrow{\textsf{ave}} \textbf{Stoch}^G_{\mathrm{det}}((X, \alpha_X), (Y, \alpha_Y))$$

Deterministic symmetrisation via averaging

Can combine averaging with stochastic symmetrisation:

$$\textbf{Stoch}(X, Y) \xrightarrow{\textsf{sym}_\gamma} \textbf{Stoch}^G((X, \alpha_X), (Y, \alpha_Y)) \xrightarrow{\textsf{ave}} \textbf{Stoch}^G_{\mathrm{det}}((X, \alpha_X), (Y, \alpha_Y))$$

When applied to a deterministic function f, the result is

$$\mathbb{E}_{\boldsymbol{G} \sim \gamma(dg|x)}[\boldsymbol{G} \cdot f(\boldsymbol{G}^{-1} \cdot x)]$$

which recovers the methods of Kim et al. [2023] and Puny et al. [2022]

Deterministic symmetrisation via averaging

Can combine averaging with stochastic symmetrisation:

$$\mathbf{Stoch}(X, Y) \xrightarrow{\mathsf{sym}_\gamma} \mathbf{Stoch}^G((X, \alpha_X), (Y, \alpha_Y)) \xrightarrow{\mathsf{ave}} \mathbf{Stoch}^G_{\mathrm{det}}((X, \alpha_X), (Y, \alpha_Y))$$

When applied to a deterministic function f, the result is

$$\mathbb{E}_{\boldsymbol{G} \sim \gamma(dg|x)}[\boldsymbol{G} \cdot f(\boldsymbol{G}^{-1} \cdot x)]$$

which recovers the methods of Kim et al. [2023] and Puny et al. [2022]

Note however that averaging is expensive, approximate, and requires convexity of Y