

Equivariant Adaptation of Large Pretrained Models

Mondal, Panigrahi, Kaba, Rajeswar, and Ravabakhsh

Presentation by Giacomo Negri

3rd of February 2026

01

Problem Statement

Context

Large Pretrained
Models achieve
SotA through
massive datasets
& training

Limitation

Sensible to OOD
samples

Problem

Redesigning and
retraining large
backbones is
expensive

Context

Large Pretrained
Models achieve
SotA through
massive datasets
& training

Limitation

Sensible to OOD
samples

Problem

Redesigning and
retraining large
backbones is
expensive

Context

Large Pretrained
Models achieve
SotA through
massive datasets
& training

Limitation

Sensible to OOD
samples

Problem

Redesigning and
retraining large
backbones is
expensive

Canonicalization Network!

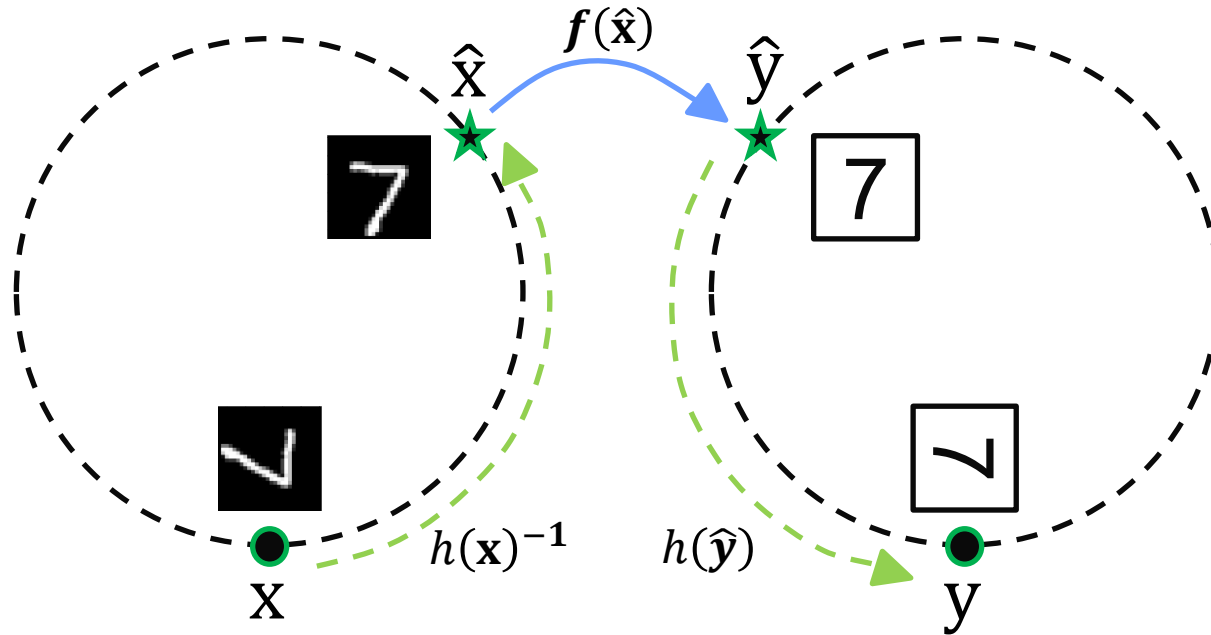
«Drop in» wrapper to confer equivariance to frozen
backbones

02

Background

You can enter a subtitle here if you need it

Single-view-plus-transformation



Group & Representation

A **group** $\mathcal{G}$ is a set equipped with associative binary operation s.t. is closed under this operation and each element $g \in \mathcal{G}$ has a unique inverse.

A **group representation** is a mapping from a group $\mathcal{G}$ to invertible linear operators:

$$\rho: \mathcal{G} \rightarrow \text{GL}(\mathcal{X})$$

the linear action is: $\rho(g) \cdot \mathbf{x}$, for each $\mathbf{x} \in \mathcal{X}$.

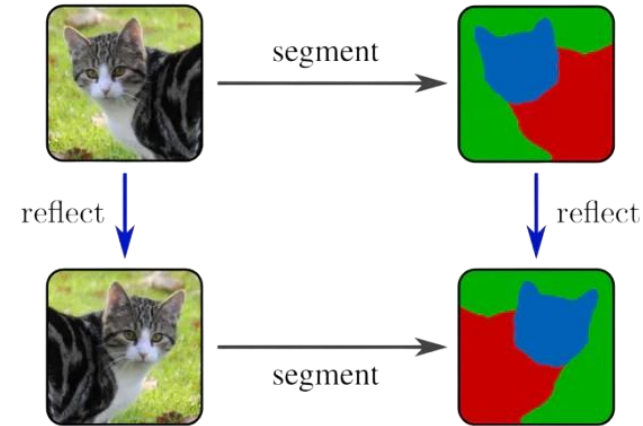
Equivariance & Invariance

A function $f: \mathcal{X} \rightarrow \mathcal{Y}$ is **equivariant** if:

$$f(\rho(g) \cdot \mathbf{x}) = \rho'(g) \cdot f(\mathbf{x})$$

where $\rho': G \rightarrow T'$ is another representation that acts on $\mathcal{Y}$.

If $\rho'(g) = I, \forall g \in \mathcal{G}$ then f is **invariant**.

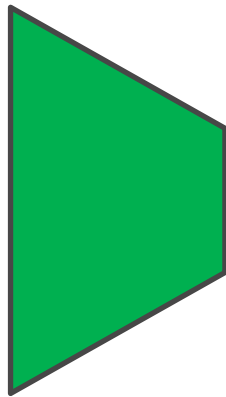


Equivariance via Canonicalization

$$f(\mathbf{x}) = c(\mathbf{x})$$

Canonicalization function: $c: \mathcal{X} \rightarrow \mathcal{G}$

$\mathbf{x} =$



$c(\mathbf{x}) = 180^\circ$ rotation

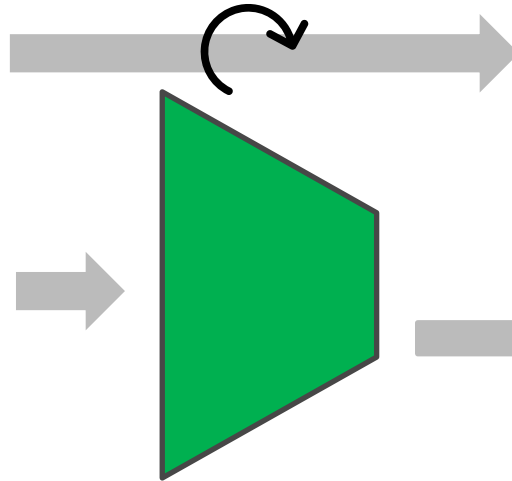
Equivariance via Canonicalization

$$f(\mathbf{x}) =$$

$$\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x}$$

1. Canonicalize

$\mathbf{x} =$



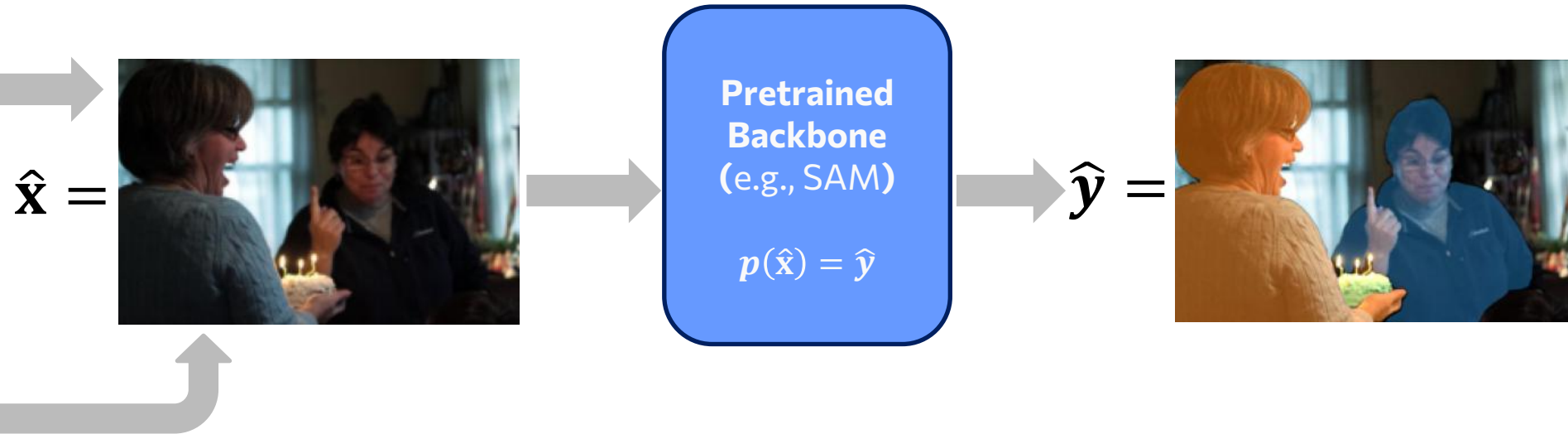
$\hat{\mathbf{x}} =$



Equivariance via Canonicalization

$$f(\mathbf{x}) = p(\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x})$$

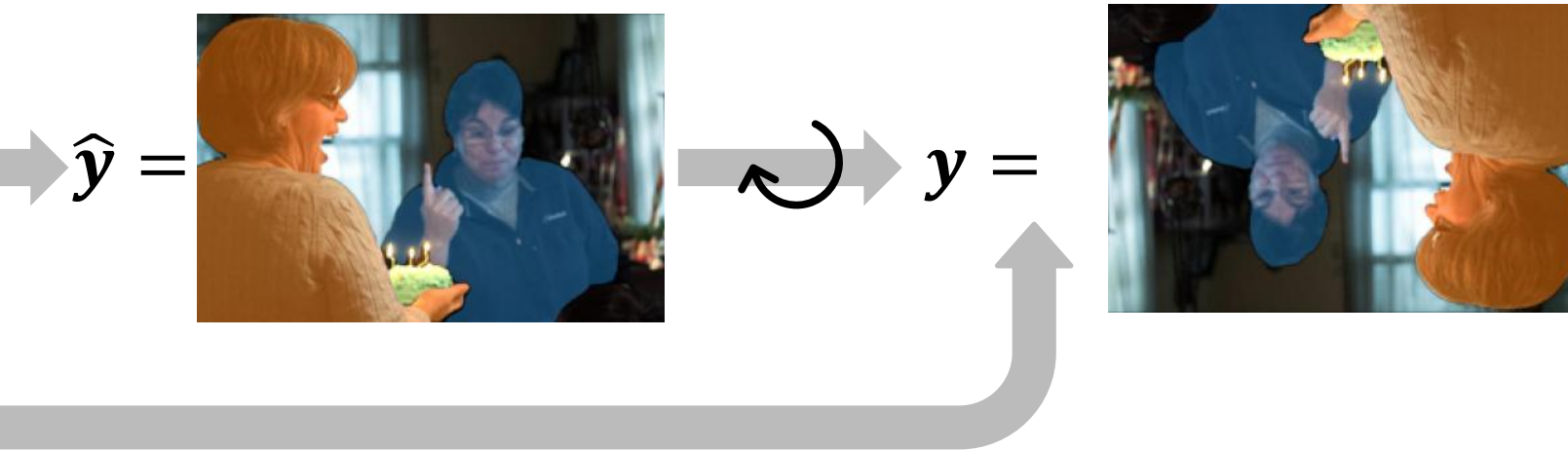
2. Predict



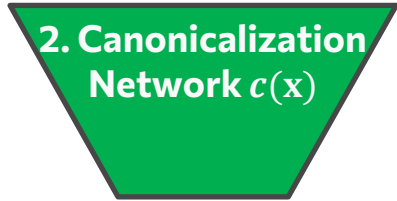
Equivariance via Canonicalization

$$f(\mathbf{x}) = \boxed{\rho'(c(\mathbf{x}))} \cdot p(\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x})$$

3. Restore



1. Input ($\mathbf{x}$)



3. Canonicalize $\hat{\mathbf{x}} = \rho(c(\mathbf{x})^{-1} \cdot \mathbf{x})$

4. Canonicalized Image ($\hat{\mathbf{x}}$)



5. Pretrained Backbone (e.g., SAM)

$$p(\hat{\mathbf{x}}) = \hat{\mathbf{y}}$$

6. Canonical Prediction ($\hat{\mathbf{y}}$)



7. Restore $\mathbf{y} = \rho'(c(\mathbf{x})) \cdot \hat{\mathbf{y}}$

7. Final Output ($\mathbf{y}$)



Equivariance via Canonicalization

$$f(\mathbf{x}) = \boxed{\rho'(c(\mathbf{x}))} \cdot p(\boxed{\rho(c(\mathbf{x})^{-1}) \cdot \mathbf{x}})$$

Canonicalize



Predict



Restore

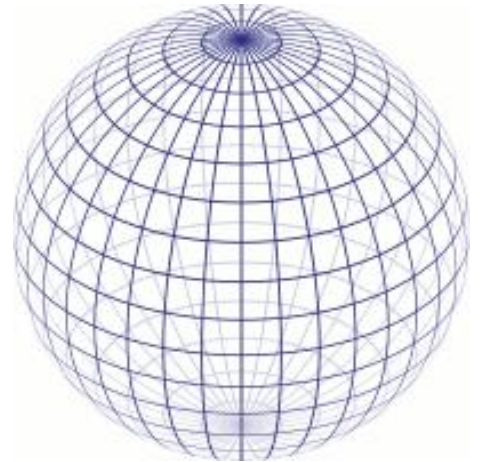
f is equivariant as long as **$c(\mathbf{x})$ is $\mathcal{G}$ –equivariant**
$$c(\rho(g) \cdot \mathbf{x}) = \rho(g) \cdot c(\mathbf{x})$$

Relaxed Equivariance

If $\mathbf{x}$ has a non-trivial stabilizer:

$$\mathcal{G}_{\mathbf{x}} = \{g \in \mathcal{G} : \rho(g) \cdot \mathbf{x} = \mathbf{x}\}$$

there is **no unique canonical representation.**



Relaxed Equivariance

If $\mathbf{x}$ has a non-trivial stabilizer:

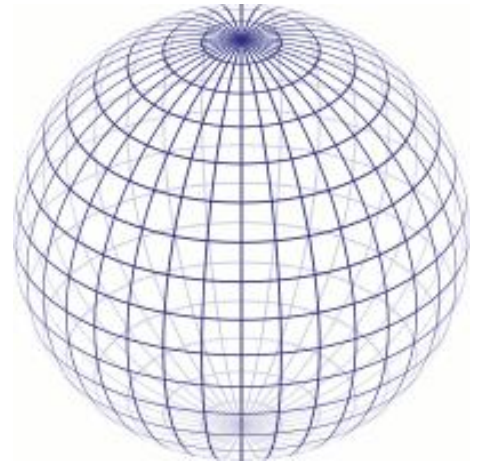
$$\mathcal{G}_{\mathbf{x}} = \{g \in \mathcal{G} : \rho(g) \cdot \mathbf{x} = \mathbf{x}\}$$

there is **no unique canonical representation.**

Equivariance is required **only up to stabilizer coset**:

$$\forall g_1, \mathbf{x} \quad \exists g_2 \in g_1 \mathcal{G}_{\mathbf{x}} : c(\rho(g_1) \cdot \mathbf{x}) = \rho'(g_2) \cdot c(\mathbf{x})$$

we can produce asymmetric output from symmetric input.

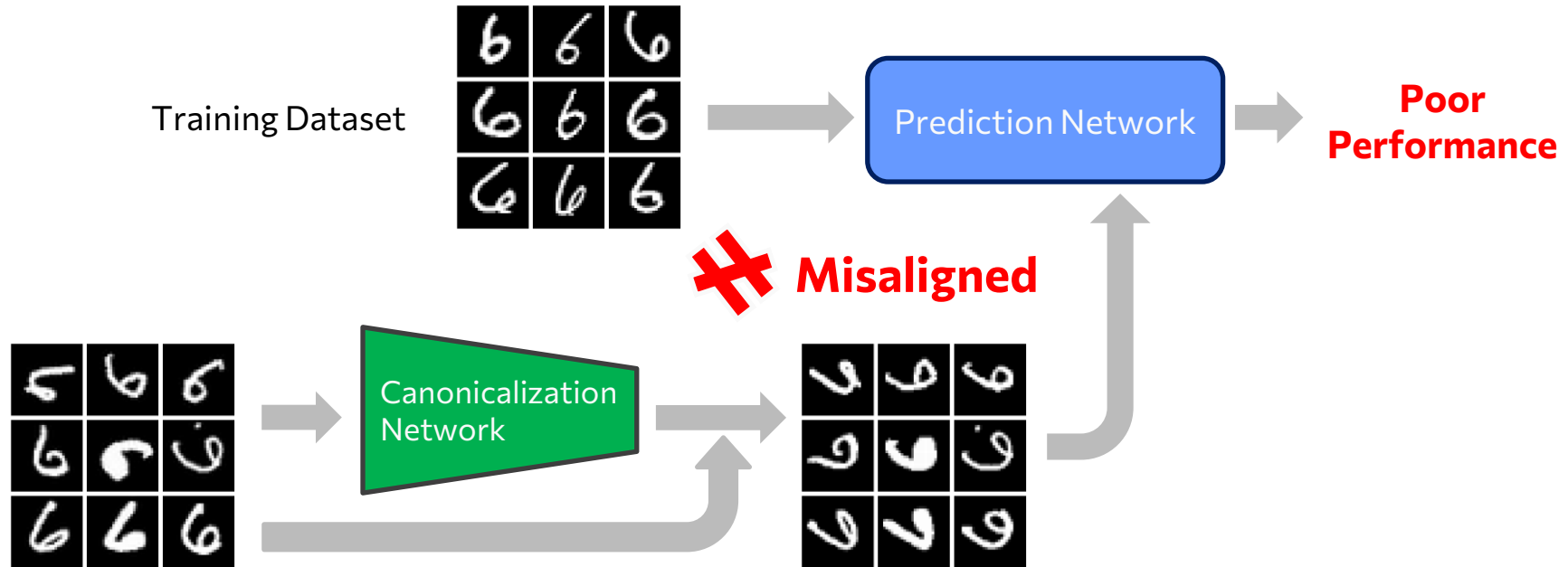


03

Learned Canonicalization Problem

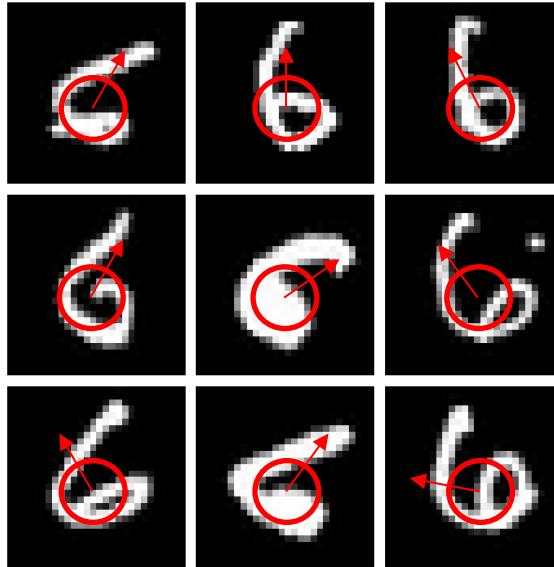
The alignment and augmentation problem

Alignment Problem

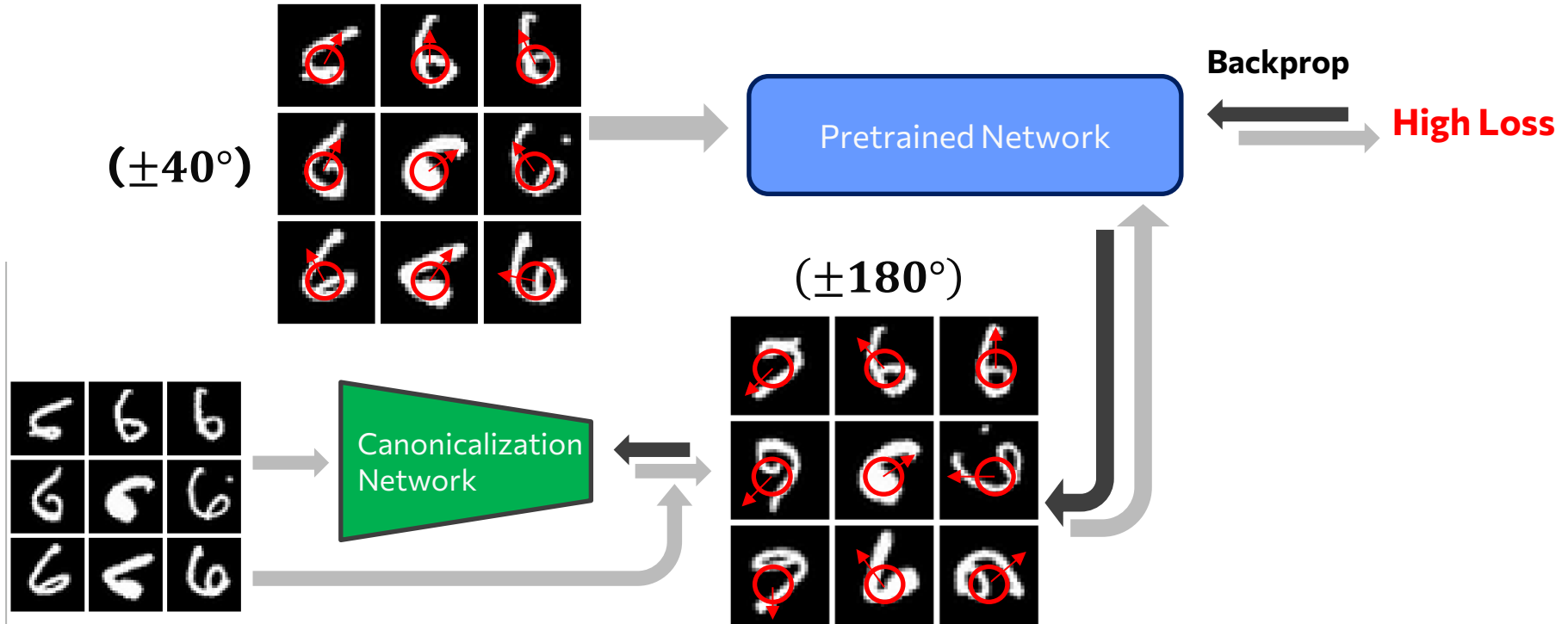


Augmentation Problem

Training Dataset ($\pm 40^\circ$)



Augmentation Problem



04

Proposed Method

Prior Regularised Learned Canonicalization

Canonicalization Prior (CP)

- **Hypothesis:** the canonical orientation should align with the distribution of the pretraining dataset $\mathcal{D}$.
- **Objective:**

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [D_{KL}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_{c(\mathbf{x})})]$$

Canonicalization Prior (CP)

- **Hypothesis:** the canonical orientation should align with the distribution of the pretraining dataset $\mathcal{D}$.
- **Objective:**

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [D_{KL}(\mathbb{P}_{\mathcal{D}} || \mathbb{P}_{c(\mathbf{x})})]$$

$$\mathcal{L}_{\text{prior}} = -\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbb{E}_{\mathbf{R} \sim q} [\log p(\mathbf{R} | c(\mathbf{x}))]$$

with $q(\mathbf{R})$ as prior and $p(\mathbf{R} | c(\mathbf{x}))$ as probability distribution of rotations

Implementation – Discrete Rotations

- 1 **Represent rotations as a finite group C_n** from which to choose from.
- 2 Network **outputs logits over all group elements** ($|G|, H, W$) using a group-equivariant model (e.g. Group CNN).
- 3 **Apply spatial averaging** to collapse feature maps (H, W) into one logit per group element ($|G|$).
- 4 Convert logits to a categorical distribution via **softmax** and **penalize** probability mass away from the identity.
- 5 During **training**, select the most likely rotation with an **argmax** and use a **straight-through gradient estimator** to backpropagate.

Continuous Rotations – Matrix Fisher

(Normal) Gaussian

$$p(\mathbf{x}) = \frac{1}{2\pi^{d/2}} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\}$$

Isotropic
 $SO(n)$ Gaussian

$$p(\mathbf{R} | \hat{\mathbf{R}}, s) = \frac{1}{n(s)} \exp(s \operatorname{Tr}[\hat{\mathbf{R}}^T \mathbf{R}])$$

where $\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^T$.

Continuous Rotations – Matrix Fisher

(Normal) Gaussian

$$p(\mathbf{x}) = \frac{1}{2\pi^{d/2}} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\}$$

Isotropic
 $SO(n)$ Gaussian

$$p(\mathbf{R} | \hat{\mathbf{R}}, s) = \frac{1}{n(s)} \exp(s \operatorname{Tr}[\hat{\mathbf{R}}^T \mathbf{R}])$$

Generalizing the **isotropic** ($S = sI$) case above:

$$p(\mathbf{R} | \mathbf{F}) = \frac{1}{n(\mathbf{F})} \exp(\operatorname{Tr}[\mathbf{F}^T \mathbf{R}])$$

where $\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and $\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^T$.

Continuous Rotations – Fisher Matrix

Generalizing the **isotropic** ($S = sI$) case above:

$$p(\mathbf{R} | \mathbf{F}) = \frac{1}{n(\mathbf{F})} \exp(\text{Tr}[\mathbf{F}^T \mathbf{R}])$$

where $\mathbf{F} = \mathbf{USV}^T$ and $\hat{\mathbf{R}} = \mathbf{UV}^T$.

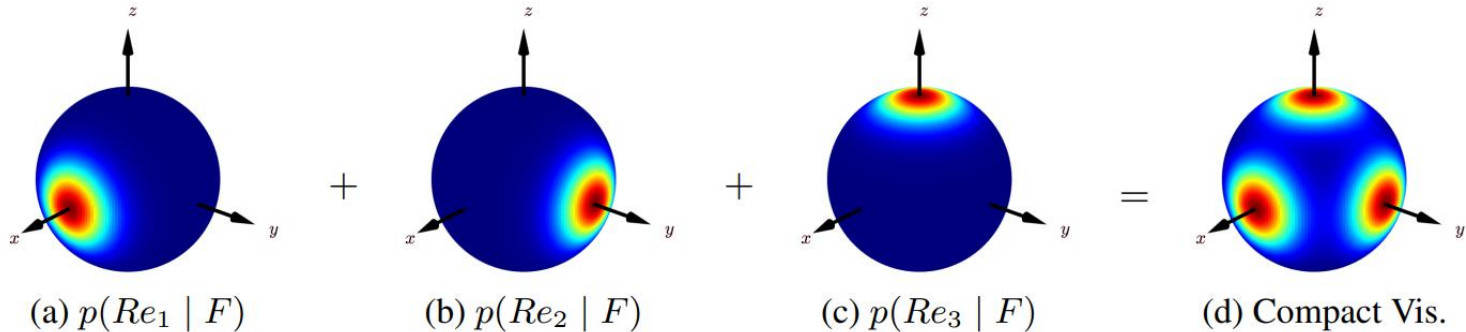
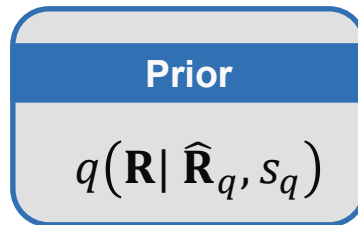
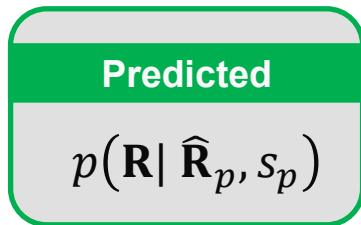


Image Source: Probabilistic orientation estimation with matrix Fisher distributions. The matrix Fisher distribution on $SO(3)$

Continuous Rotations – Prior Loss



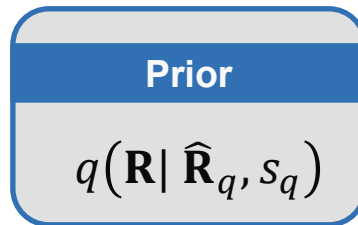
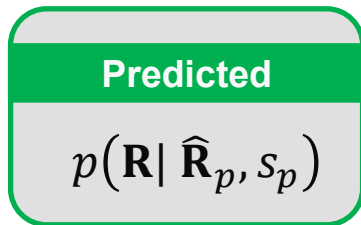
Isotropic
 $SO(n)$ Gaussian

$$p(\mathbf{R} | \hat{\mathbf{R}}, s) = \frac{1}{n(s)} \exp(s \operatorname{Tr}[\hat{\mathbf{R}}^T \mathbf{R}])$$

The cross-entropy is:

$$\mathbb{E}_{\mathbf{R} \sim q} [\log p(\mathbf{R} | \hat{\mathbf{R}}_p, s_p)] = N(s_q) s_p \operatorname{Tr}(\hat{\mathbf{R}}_p^T \hat{\mathbf{R}}_q) + \log c(s_p)$$

Continuous Rotations – Prior Loss



The cross-entropy is:

$$\mathbb{E}_{\mathbf{R} \sim q} [\log p(\mathbf{R} | \hat{\mathbf{R}}_p, s_p)] = N(s_q) s_p \text{Tr}(\hat{\mathbf{R}}_p^T \hat{\mathbf{R}}_q) + \log c(s_p)$$

defining $\lambda = N(s_q) s_p$ and setting the prior $\hat{\mathbf{R}}_q = \mathbf{I}$:

$$\mathcal{L}_{\text{prior}} = -\lambda \text{Tr}(\hat{\mathbf{R}}_{c(x)}) = -\frac{\lambda}{2} \|\hat{\mathbf{R}}_{c(x)} - \mathbf{I}\|_F$$

Implementation-Continuous Rotations

- ① **Feed the inputs** $(B, 3, H, W)$ into the canonicalization network, E(2)-Steerable Network (or PointNet/DGCCN)
- ② Network returns **two vector fields** $(B, 2, H, W)$.
- ③ The fields are **spatially averaged** $(B, 2)$.
- ④ The **Gram-Schmidt orthonormalization** $(B, 2, 2)$ is applied to these vectors to guarantee the output is a valid rotation matrix $R \in SO(2)$ (or $SO(3)$).
- ⑤ Use the continuous prior loss $\|\mathbf{R}_{c(x)} - \mathbf{I}\|_F$ to push predicted rotations toward the identity.

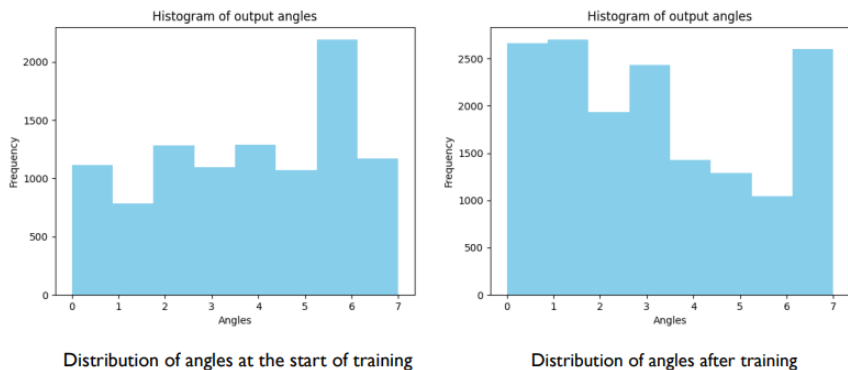
05

Experiments and Results

Identity Mapping

		Training Completed	
Dataset ↓	Model	✗	✓
CIFAR10 [31]	Learned Canonicalization (LC) [30]	0.11	0.23
	Prior-Regularized LC (ours)	0.11	0.76

Learned Canonicalization (LC)



Prior – Regularized LC

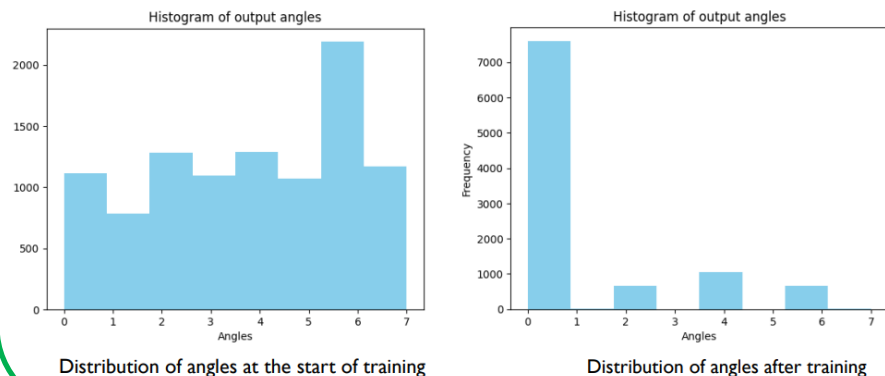


Image Classification

pretrained Large Prediction Network →		ResNet50		ViT	
Datasets ↓	Model	Acc	C8-Avg Acc	Acc	C8-Avg Acc
CIFAR10 [31]	Vanilla	96.97 ± 0.01	57.77 ± 0.25	98.13 ± 0.04	63.59 ± 0.48
	Rotation Augmentation	94.91 ± 0.07	90.11 ± 0.19	96.26 ± 0.15	93.67 ± 0.39
	Learned Canonicalization (LC) [30]	93.29 ± 0.01	92.96 ± 0.09	95.00 ± 0.01	94.80 ± 0.02
	C8-Aug.	95.76 ± 0.07	94.36 ± 0.09	96.36 ± 0.02	94.17 ± 0.08
	Prior-Regularized LC (ours)	96.19 ± 0.01	95.31 ± 0.17	96.14 ± 0.14	95.08 ± 0.10

- Large Pretrained Models are **not equivariant**
- **Augmentation reduce** performance
- **Prior Reg. LC** is the **best performer** for augmented dataset
- **Lower accuracy** for **Prior Reg. LC** w.r.t. Vanilla for **original dataset**.

Instance Segmentation

Pretrained Large Segmentation Network →		MaskRCNN (46.4 M)		SAM (641 M)	
Datasets ↓	Model	mAP	C4-Avg mAP	mAP	C4-Avg mAP
COCO [22]	Zero-shot (0 M)	45.57	27.67	62.34	58.78
	Prior-Regularized LC [†] (0.2 M)	35.77	35.77	59.28	59.28
	Prior-Regularized LC [‡] (1.9 M)	44.51	44.50	62.13	62.13

- Prior-Regularized LC enforces exact C4-equivariance for both models without hurting original performance.
- **More expressive canonicalization** networks **preserve accuracy better** while still guaranteeing equivariance.

Point Clouds

Prior-Regularized LC
improves robustness to
arbitrary $SO(3)$ rotations

**Avoiding large rotation
augmentations** on the
prediction model removes
key bottlenecks for
learned canonicalization.

Task →	Classification			Part Segmentation	
Dataset →	ModelNet40			ShapeNet	
Method ↓	z/z	$z/SO(3)$	$SO(3)/SO(3)$	$z/SO(3)$	$SO(3)/SO(3)$
PointNet [3]	85.9	19.6	74.7	38.0	62.3
DGCNN [48]	90.3	33.8	88.6	49.3	78.6
VN-PointNet	77.5	77.5	77.2	72.4	72.8
VN-DGCNN	89.5	89.5	90.2	81.4	81.4
LC-PointNet[30]	79.9 ± 1.3	79.6 ± 1.3	79.7 ± 1.3	73.5 ± 0.8	73.6 ± 1.1
LC-DGCNN[30]	88.7 ± 1.8	88.8 ± 1.9	90.0 ± 1.1	78.4 ± 1.0	78.5 ± 0.9
Ours (with pretrained PointNet and DGCNN for each task)					
	no-aug/ z	no-aug/ $SO(3)$		no-aug/ $SO(3)$	
PRLC-PointNet	84.1 ± 1.1	84.3 ± 1.2		82.6 ± 1.3	
PRLC-DGCNN	90.2 ± 1.4	90.2 ± 1.3		84.3 ± 0.8	

06

Conclusion

Advantages



Enables backbone scalability



Versatile to diverse backbone architectures



Efficient, for reported task (+ 7.3% inference | +0.3% params, on SAM)



Prior Regularized Learned Canonicalization improved performance consistently

Limitations



Restricted range of transformation



Limited expressivity of the canonicalization function.



Continuous rotations while theoretically sounding remain difficult to optimize in the image domain.



Canonicalization quality directly influences downstream performance.

Critical Observation



Circular dependency on Equivariant Network



Limited inference time reporting



Equivariant Network training time (more/less efficient than FA)

Thank you!

Any questions?