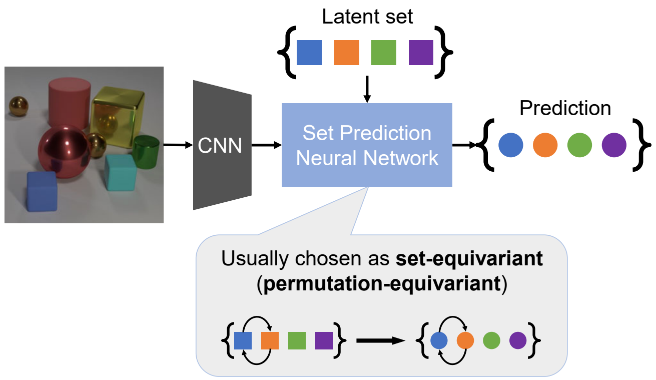
Multiset-Equivariant Set Prediction with Approximate Implicit Differentiation

Yan Zhang*[1], **David W. Zhang***[2], Simon Lacoste-Julien[134], Gertjan J. Burghouts[5], Cees G. M. Snoek[2]
[1]Samsung – SAIT AI Lab, Montreal [2]University of Amsterdam [3]Mila, Université de Montreal [4]Canada CIFAR AI Chair [5]TNO *equal contribution

Equivariance: Role in Set Prediction



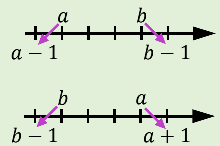
Limitation of Set-Equivariance

Q: Can set-equivariant neural networks represent `push_apart`?

Definition of `push_apart`$([a, b])$:
Push smaller to the left and larger to the right.

If $a \leq b$

Else



Examples:

`push_apart`$([1,2]) = [0,3]$

`push_apart`$([2,1]) = [3,0]$

Looks like it's set-equivariant, *but what if* $a = b$?

`push_apart`$([1,1]) = [0,2]$

`push_apart`$([1,1]) = [0,2]$

A: Set-equivariant functions **cannot represent** `push_apart`!

Multiset-equivariance without set-equivariance is the proper permutation symmetry for set prediction and enables **separating equal elements**.

Video, code, pre-trained models and run statistics available at
`https://github.com/davzha/multiset-equivariance`

Multiset-Equivariance

Previous notion of permutation equivariance:
A function $f: \mathbb{R}^{n \times d_1} \rightarrow \mathbb{R}^{n \times d_2}$ is **set-equivariant** iff for any permutation matrix $\boldsymbol{P}$:

$$f(\boldsymbol{PX}) = \boldsymbol{P} f(\boldsymbol{X})$$

Better:
A function $f: \mathbb{R}^{n \times d_1} \rightarrow \mathbb{R}^{n \times d_2}$ is **multiset-equivariant** iff for any permutation matrix $\boldsymbol{P}_1$, there exists $\boldsymbol{P}_2$ such that $\boldsymbol{P}_1\boldsymbol{X} = \boldsymbol{P}_2\boldsymbol{X}$:

$$f(\boldsymbol{P}_1\boldsymbol{X}) = \boldsymbol{P}_2 f(\boldsymbol{X})$$

Implicit Deep Set Prediction Networks

$$\texttt{iDSPN}(\boldsymbol{z}) = \arg\min_{\boldsymbol{Y}} L(\boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\theta})$$

Forward-pass:
Find the best output set $\boldsymbol{Y}$, by minimizing $L(\boldsymbol{Y}, \boldsymbol{z}, \boldsymbol{\theta}) = \|g(\boldsymbol{Y}, \boldsymbol{\theta}) - \boldsymbol{z}\|$ via gradient descent.

Backward-pass:
Calculate gradients for $\boldsymbol{\Theta} = (\boldsymbol{z}, \boldsymbol{\theta})$ using (approximate) implicit differentiation. In detail, the optimality condition is:

$$\nabla_{\boldsymbol{Y}} L(\boldsymbol{Y}^*, \boldsymbol{\theta}) = 0$$

We differentiate both sides to get a linear system $\boldsymbol{HX} = \boldsymbol{B}$:

$$\frac{\partial \nabla_{\boldsymbol{Y}} L(\boldsymbol{Y}^*, \boldsymbol{\Theta})}{\partial \boldsymbol{Y}^*} \frac{\partial \boldsymbol{Y}^*(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = -\frac{\partial \nabla_{\boldsymbol{Y}} L(\boldsymbol{Y}^*, \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}}$$

Solving the linear system for $\boldsymbol{X}$ gives us the Jacobian for backprop.

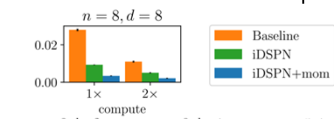
Multiset-equivariance more powerful than set-equivariance

Example: Given $[a, a, b, b, b]$, number each letter separately: $[(a,0),(a,1),(b,0),(b,1),(b,2)]$.

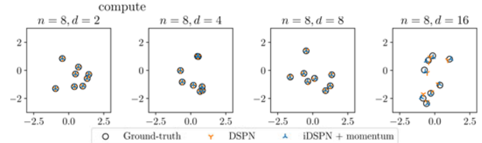
Model	Set-equivariant	Multiset-equivariant	Training samples 1×	10×	100×
DeepSets	✓	✓	$0.0_{\pm 0}$	$0.0_{\pm 0}$	$0.0_{\pm 0}$
Transformer without PE	✓	✓	$0.0_{\pm 0}$	$0.0_{\pm 0}$	$0.0_{\pm 0}$
Transformer with random PE	✗	✗	$28.5_{\pm 8.1}$	$22.6_{\pm 5.2}$	$22.7_{\pm 7.7}$
Transformer with PE	✗	✗	$0.0_{\pm 0}$	$46.4_{\pm 26.2}$	$94.4_{\pm 1.8}$
BiLSTM	✗	✗	$0.0_{\pm 0}$	$25.1_{\pm 20.2}$	$\mathbf{97.9}_{\pm 2.1}$
DSPN	✗	✓	$89.4_{\pm 2.4}$	$88.9_{\pm 4.9}$	$90.8_{\pm 4.2}$
iDSPN (ours)	✗	✓	$\mathbf{96.8}_{\pm 0.6}$	$\mathbf{98.1}_{\pm 0.3}$	$\mathbf{97.9}_{\pm 0.6}$

Implicit DSPN greatly improves over baseline DSPN

Tasks: Auto-encode random d-dimensional point sets of size n.



Examples:



Much more precise than SotA on CLEVR Object Prediction

Model	Image size	AP_{∞}	AP_1	$AP_{0.5}$	$AP_{0.25}$	$AP_{0.125}$	$AP_{0.0625}$
Slot MLP	128x128	$19.8_{\pm 1.6}$	$1.4_{\pm 0.3}$	$0.3_{\pm 0.2}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	—
DSPN	128x128	$85.2_{\pm 4.8}$	$81.1_{\pm 5.2}$	$47.4_{\pm 17.6}$	$10.8_{\pm 9.0}$	$0.6_{\pm 0.7}$	—
Slot Attention	128x128	$94.3_{\pm 1.1}$	$86.7_{\pm 1.4}$	$56.0_{\pm 3.6}$	$10.8_{\pm 1.7}$	$0.9_{\pm 0.2}$	—
Slot Attention*	128x128	$96.4_{\pm 0.5}$	$93.6_{\pm 0.8}$	$80.4_{\pm 2.1}$	$26.5_{\pm 2.7}$	$2.6_{\pm 0.3}$	—
Slot Attention†	128x128	$90.6_{\pm 1.8}$	$89.1_{\pm 2.1}$	$84.4_{\pm 2.5}$	$50.3_{\pm 3.8}$	$7.9_{\pm 1.0}$	—
iDSPN	128x128	$\mathbf{98.8}_{\pm 0.5}$	$\mathbf{98.5}_{\pm 0.6}$	$\mathbf{98.2}_{\pm 0.6}$	$\mathbf{95.8}_{\pm 0.7}$	$\mathbf{76.9}_{\pm 2.5}$	$\mathbf{32.3}_{\pm 3.9}$
iDSPN	256x256	*99.4*$_{\pm 0.3}$	*99.2*$_{\pm 0.4}$	*99.0*$_{\pm 0.5}$	*97.8*$_{\pm 0.7}$	*86.9*$_{\pm 1.0}$	*47.2*$_{\pm 3.8}$