Adaptive Robust Pooling and Feature Projections in Deep Declarative Networks

Wenbo Du
Supervisor: Professor Stephen Gould

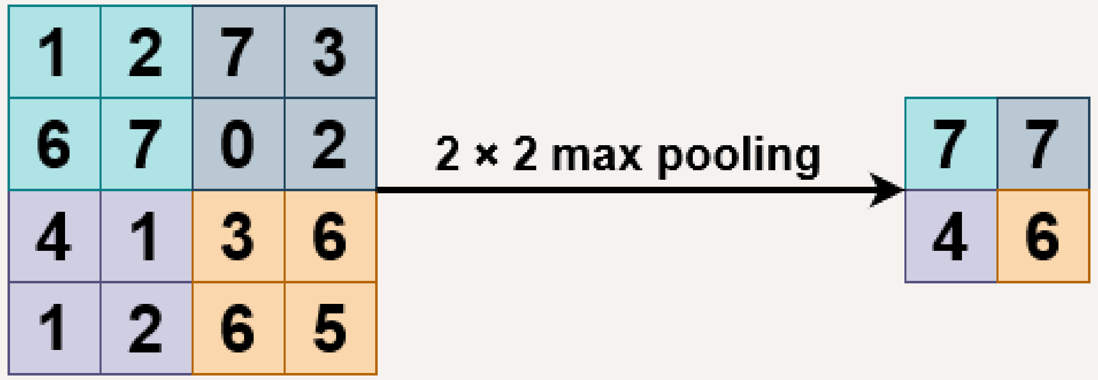
Outlines

- Background
- Adaptive Robust Pooling
- Adaptive Feature Projections
- Conclusion & Future Work

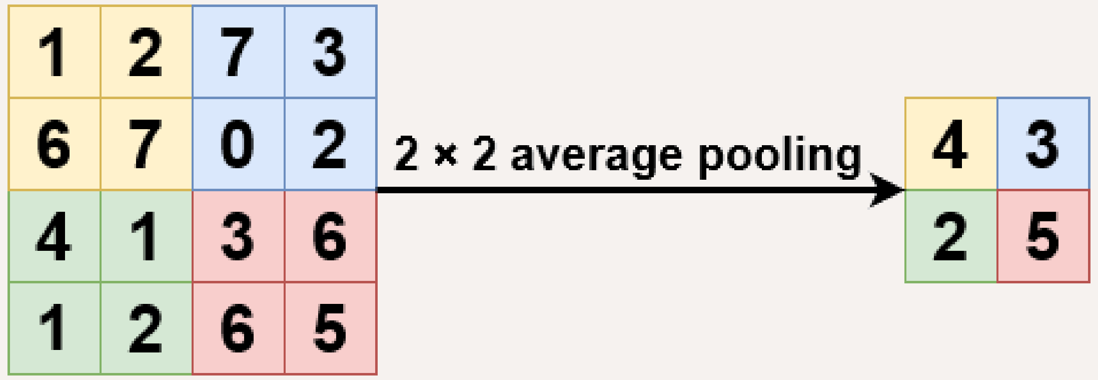
Background: Pooling in Convolutional Neural Networks

By algorithms:

- Max pooling



- Average pooling

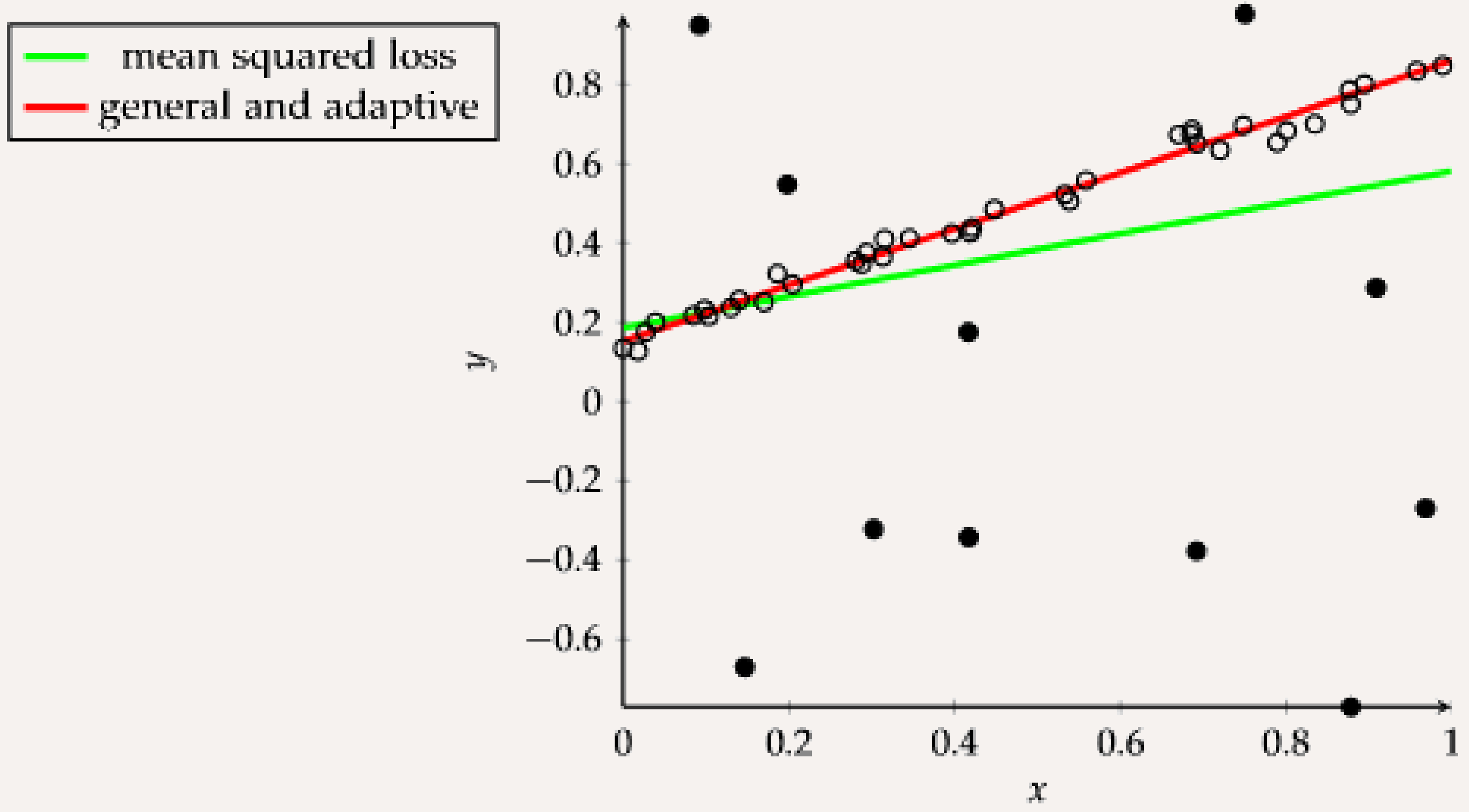


By operation region:

- Local pooling:
  one value for each patch of features.
- Global pooling:
  a single value for all features.

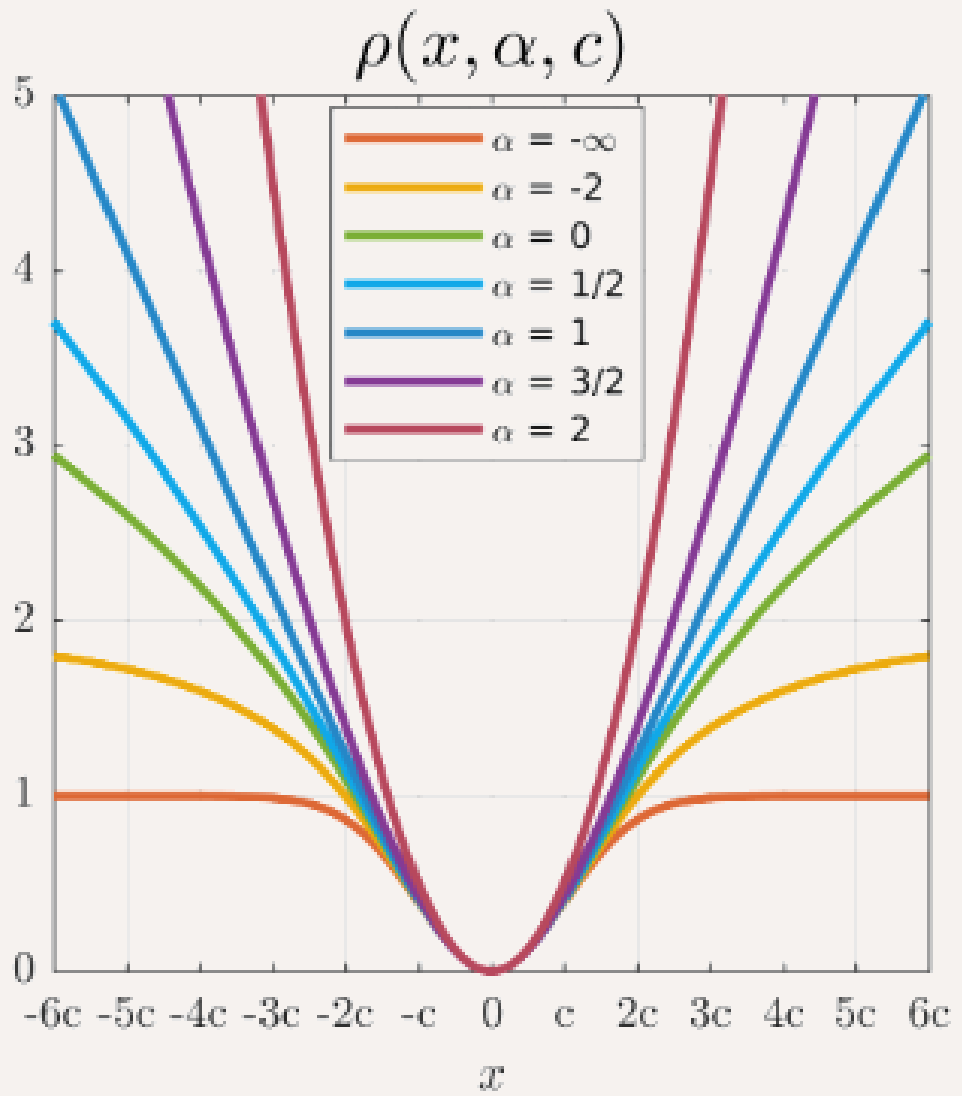
Pooling in this project refers to global pooling.

Background: A General and Adaptive Robust Loss Function



Robustness: a model's sensitivity to outliers.

Robust loss function: loss functions that are less influenced by outliers.

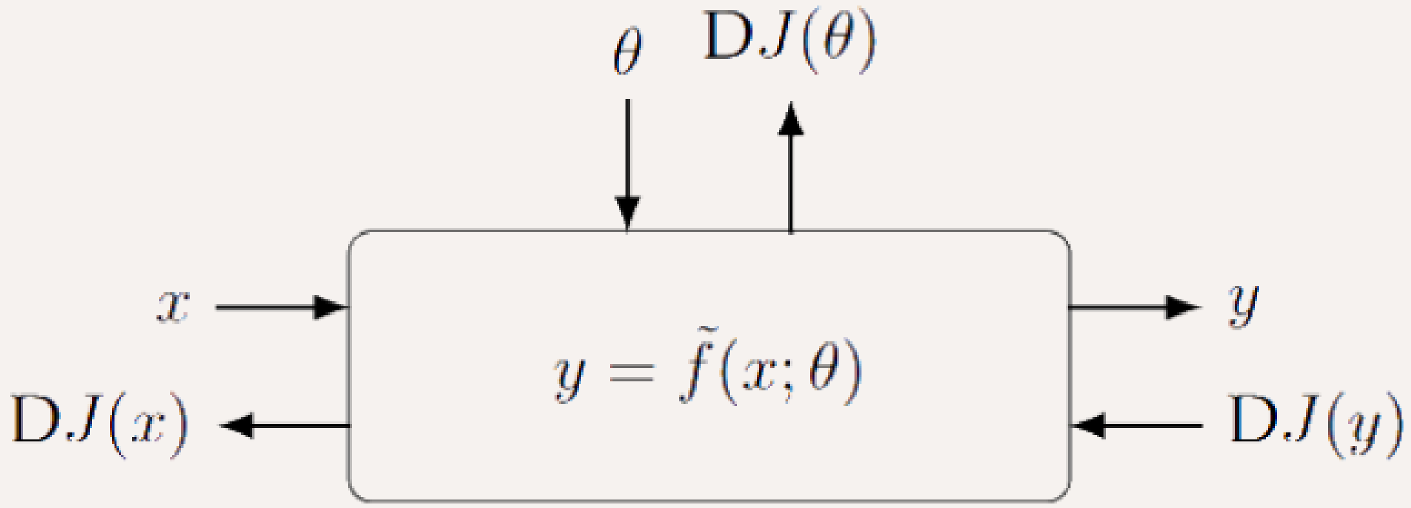
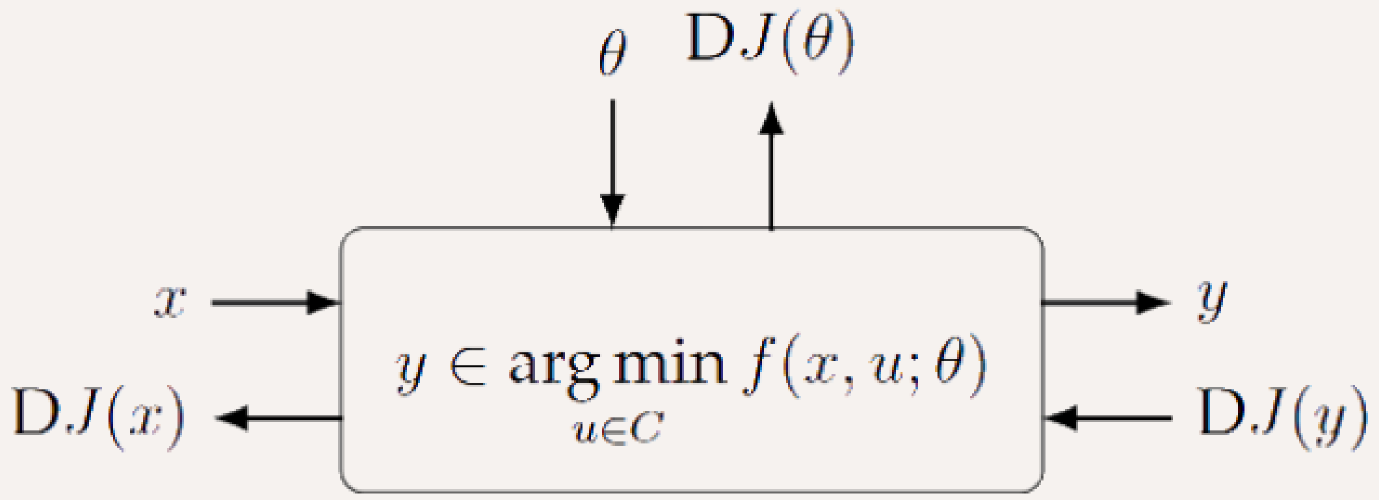


Function definition:

$$f(x, \alpha, c) = \frac{|\alpha - 2|}{\alpha} \left(\left(\frac{(x/c)^2}{|\alpha - 2|} + 1 \right)^{\alpha/2} - 1 \right)$$

- Generalise many existing loss functions with different α.
- Gradient-based learning for α (robustness) and c (outlier threshold).

Background: Deep Declarative Networks (DDN)

	Traditional Neural Network Nodes	**Deep Declarative Networks Nodes**
		
Forward	Explicitly defined function eg. ReLU: y= max(0, x)	Implicitly defined argmin problem eg. $y \in \operatorname{arg\,min}_{u \in \mathbb{R}^m} f(x, u)$ subject to $h_i(x, u) = 0, \quad i = 1, \ldots, p$
Backward	Explicit differentiation	Implicit function theorem

Two types of network nodes can co-exist in a network.

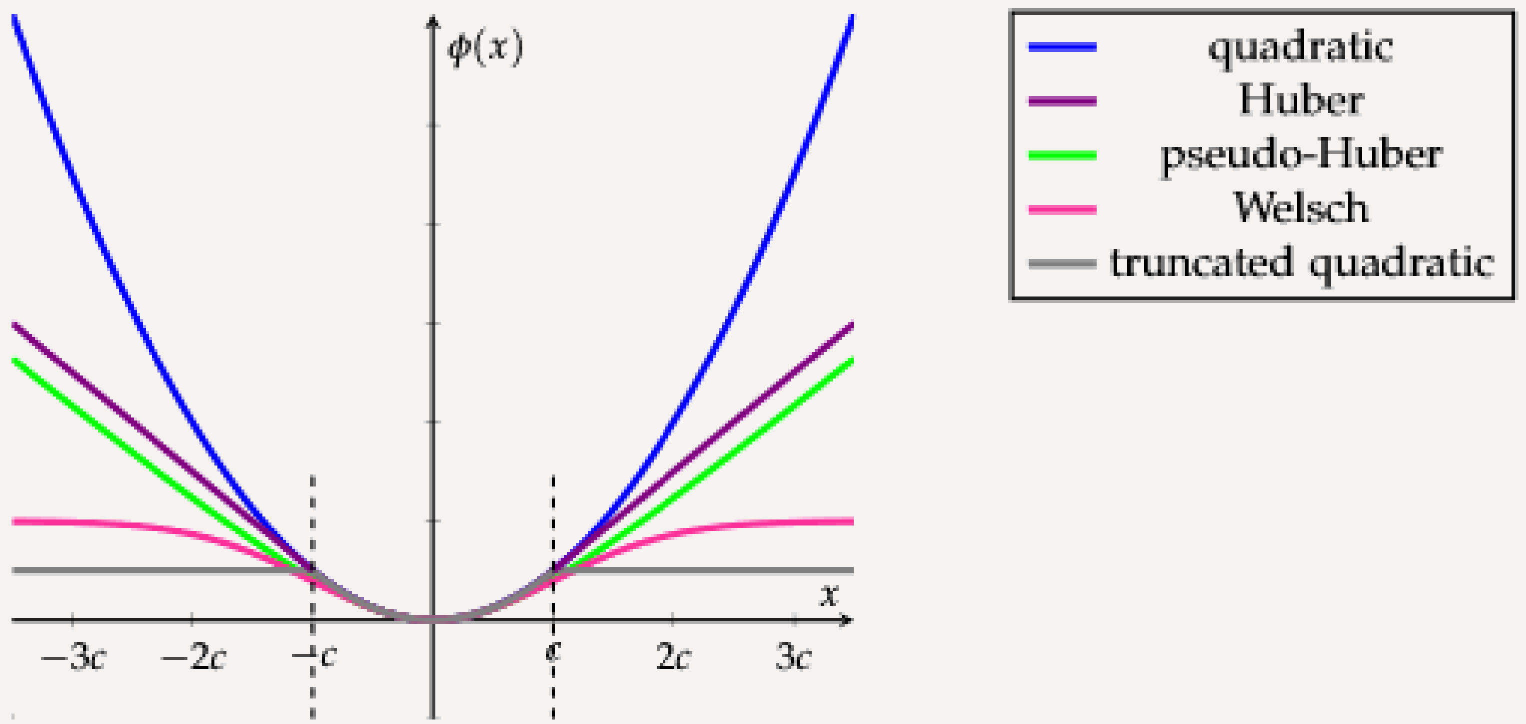
Adaptive Robust Pooling: Previous Work

- Robust pooling, as an instance of DDN, is more robust to outliers than max pooling and average pooling.

Forward $y \in \arg\min_{u \in \mathbb{R}} \sum_{i=1}^{n} \phi(u - x_i, c)$

Backward Dy(x)

- Outlier threshold *c* in robust pooling is a predetermined constant.
- ϕ can be one of loss functions:



***c* needs to be searched manually in new models/datasets.**

Adaptive Robust Pooling: My Work

	Adaptive robust pooling	**General and adaptive robust pooling**
Forward	$y \in \arg\min_{u \in \mathbb{R}} \sum_{i=1}^{n} \phi(u - x_i, c)$ subject to $c > 0$	$y \in \arg\min_{u \in \mathbb{R}} \sum_{i=1}^{n} \frac{\lvert\alpha - 2\rvert}{\alpha}\left(\left(\frac{(\frac{u - x_i}{c})^2}{\lvert\alpha - 2\rvert} + 1\right)^{\frac{\alpha}{2}} - 1\right)$ subject to $c > 0$ $0 \leq \alpha \leq 3$
Backward	Dy(x), Dy(c)	Dy(x), Dy(c), Dy(α)

Gradient calculations use DDN theory.

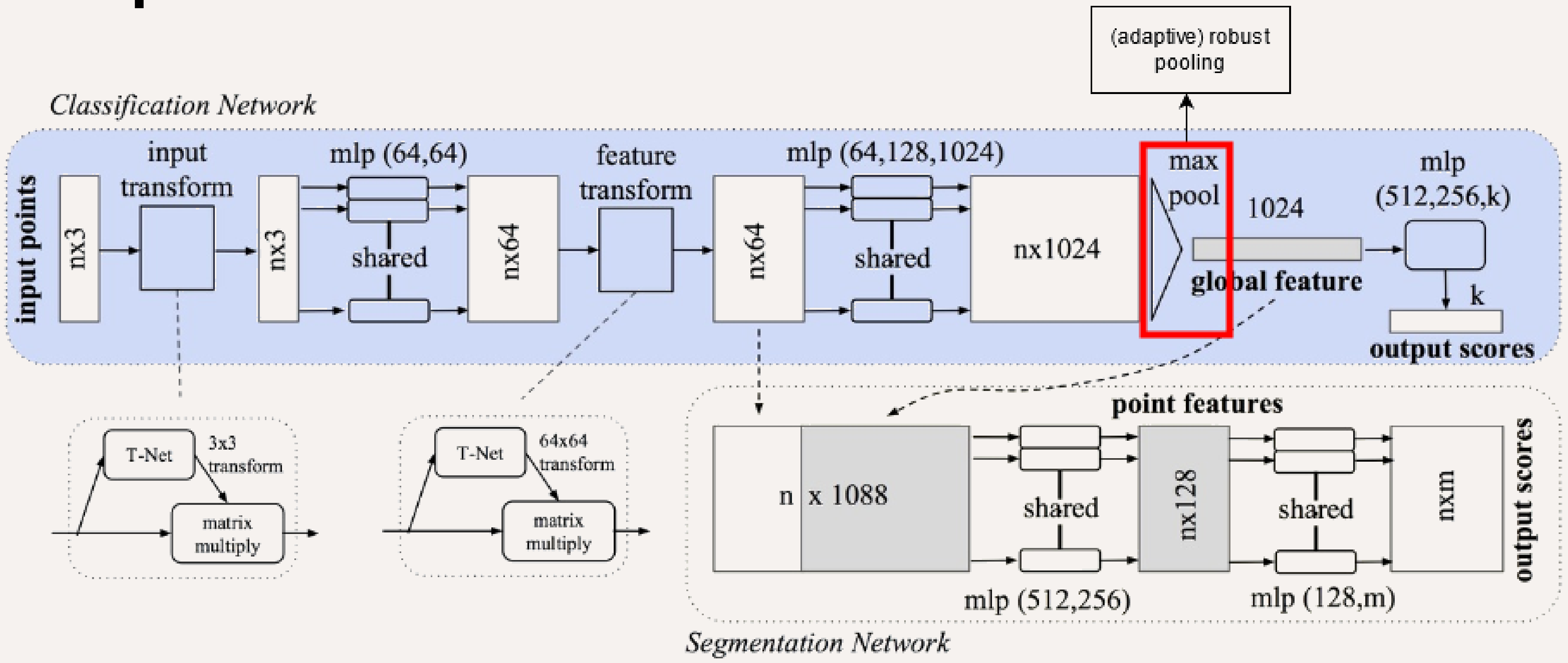
c and α are updated in end-to-end backpropagation in networks.

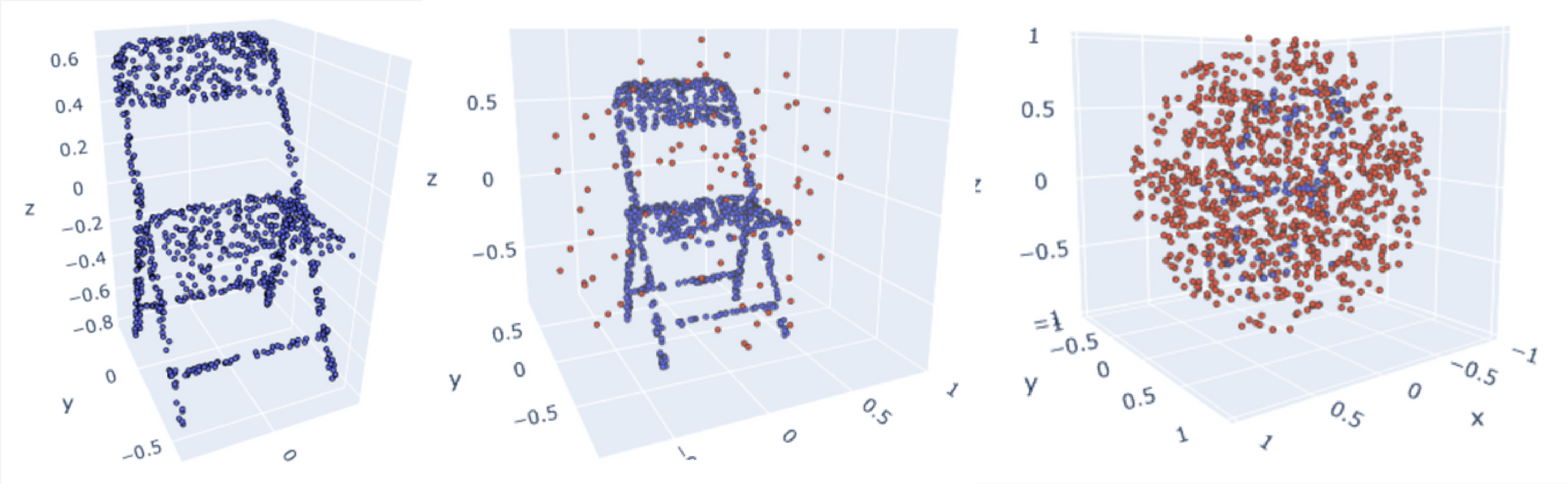
Adaptive Robust Pooling: Experiments

- Point cloud classification with PointNet and ModelNet40.

- Robust pooling with constant *c* as baselines [Gould, 2020].

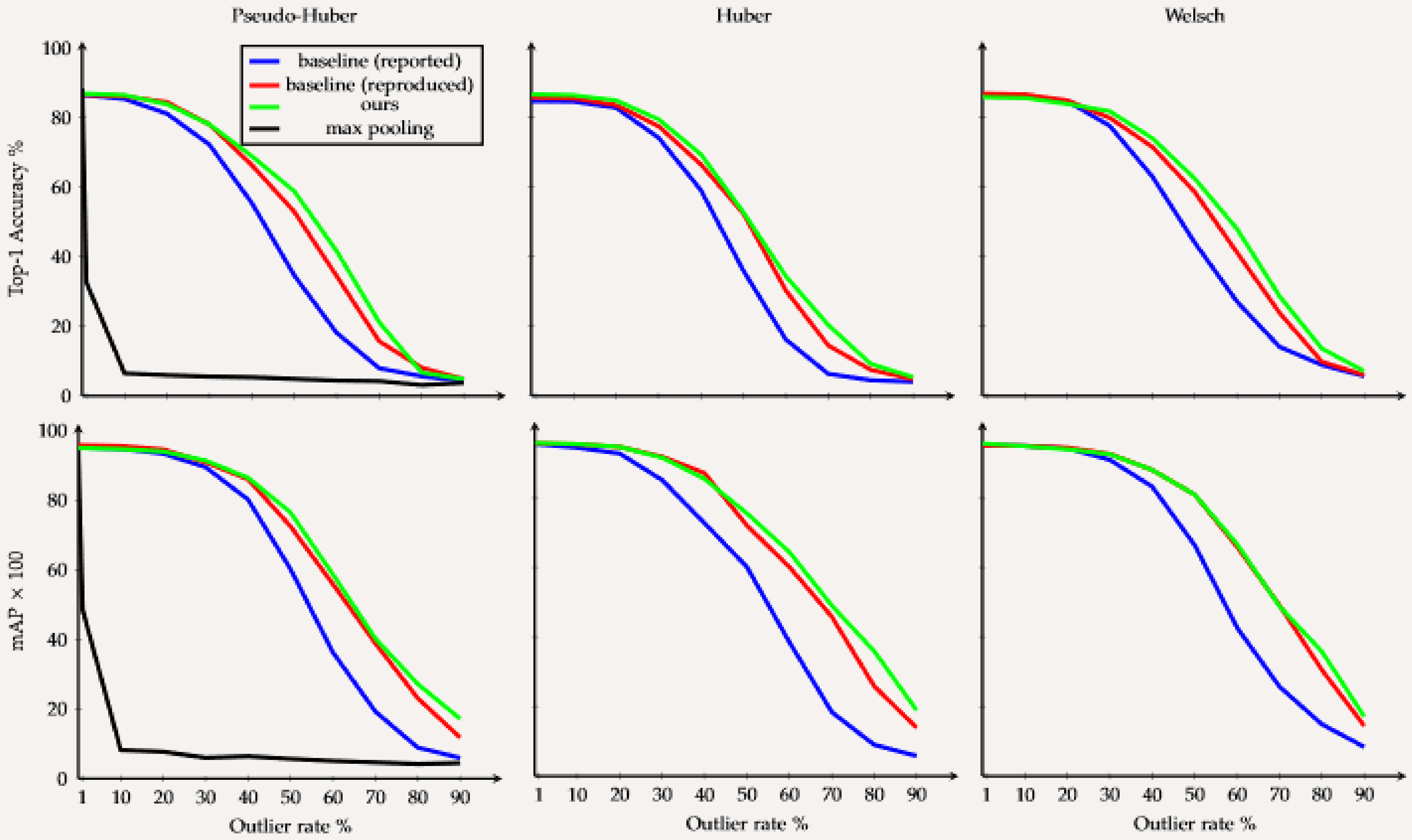
- Outliers are added to point clouds for robustness evaluation.





Adaptive Robust Pooling: Results

Train without outliers, test with outliers.



Adaptive Robust Pooling: Summary

- Our methods outperform baselines in most cases.

- The runtime and memory increases are acceptable.

Pooling type	Forward (ms)		Backward (ms)	
	baseline	ours	baseline	ours
Pseudo-Huber	**3.75**	6.78	**0.20**	0.25
Huber	**5.63**	9.825	**0.23**	0.15
Welsch	**19.95**	24.23	**0.23**	0.28

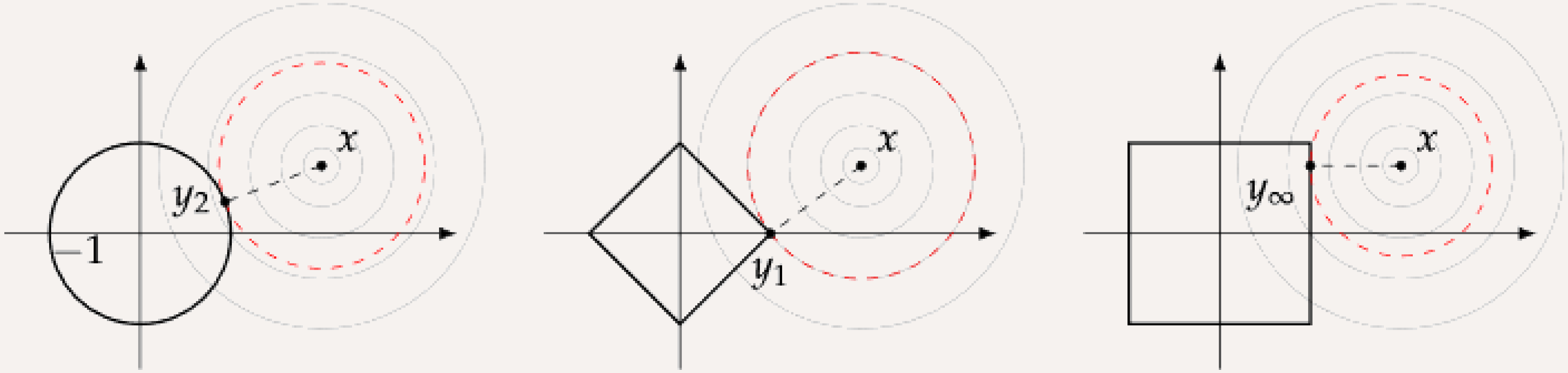
- The improvement in general and adaptive robust pooling is not significant.

Adaptive Feature Projections: Previous Work

- Euclidean projections onto L2 ball and sphere are common operations in deep learning.
- With DDN theory, we can embed projections onto L1, L∞ ball and sphere as network nodes.

	Sphere	**Ball**
Forward	$y_p \in \begin{array}{ll} \arg\min_{u \in \mathbb{R}^n} & \frac{1}{2}\|u - x\|_2^2 \\ \text{subject to} & \|u\|_p = r \end{array}$	$y_p \in \begin{array}{ll} \arg\min_{u \in \mathbb{R}^n} & \frac{1}{2}\|u - x\|_2^2 \\ \text{subject to} & \|u\|_p \leq r \end{array}$
Backward	Dy(x)	Dy(x)

- Currently, the parameter radius r in projections is set as a constant.



Hand tuning for constant r is required for different models/datasets.

Adaptive Feature Projections: My Work

- We introduce gradient-based learning for r.

	Sphere	**Ball**
Forward	$y_p \in \arg\min_{u \in \mathbb{R}^n} \frac{1}{2}\|u - x\|_2^2$ subject to $\|u\|_p = r$, $r > 0$	$y_p \in \arg\min_{u \in \mathbb{R}^n} \frac{1}{2}\|u - x\|_2^2$ subject to $\|u\|_p \leq r$, $r > 0$
Backward	Dy(x), Dy(r)	Dy(x), Dy(r)

Gradient calculations use DDN theory.

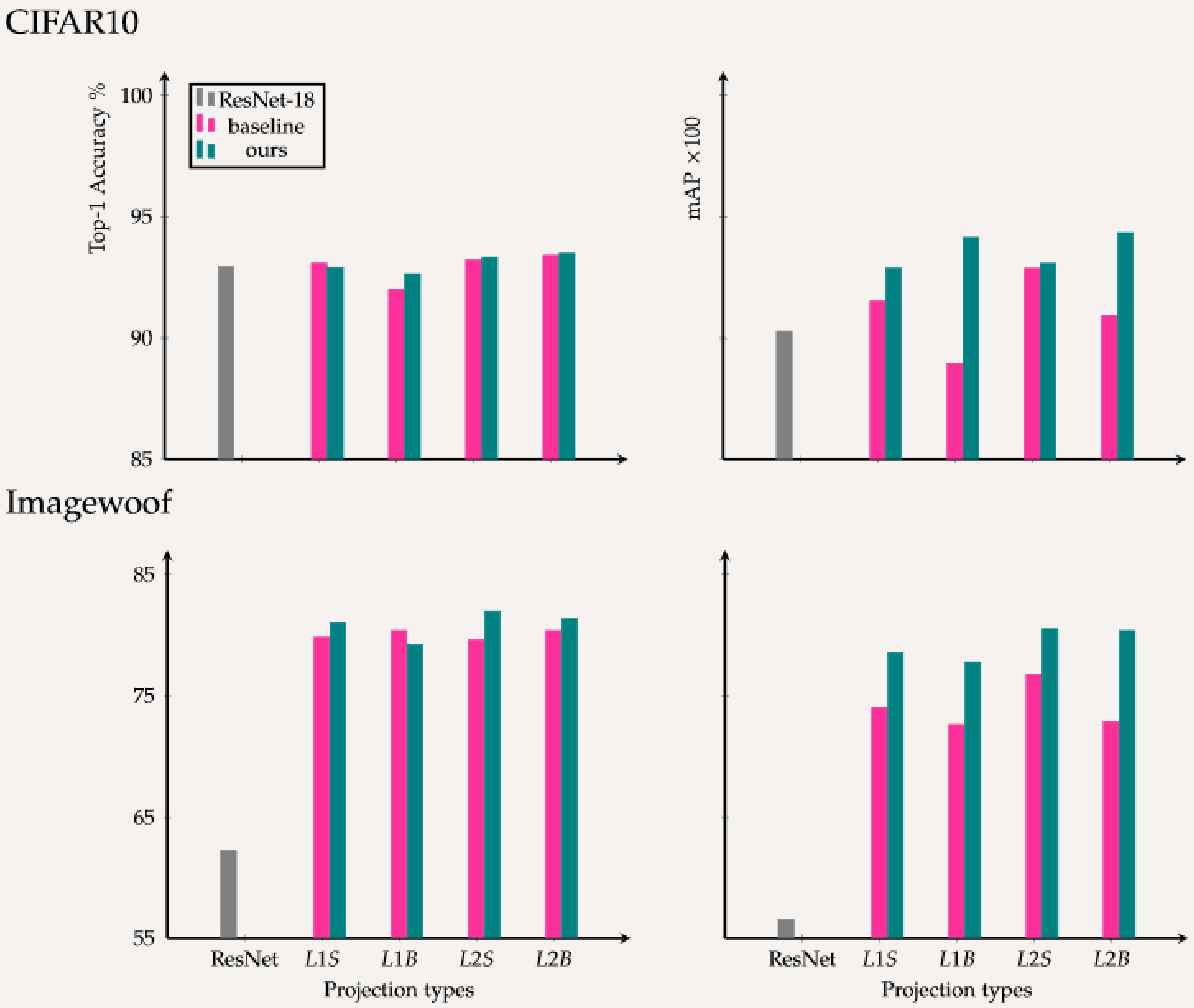
Now, parameter r is updated within networks.

Adaptive Feature Projections: Experiments

- Prior work has shown feature projections increase mean average precision significantly.
- Image classification with several convolutional neural networks: eg. ResNet18.
- Two datasets: CIFAR10 and Imagewoof (10-class subset of ImageNet)
- Original networks and projection-inserted networks are set as baselines.

layer name	output size	ResNet-18
conv1	112×112×64	7×7, 64, stride 2
conv2_x	56×56×64	3×3 max pooling, stride 2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$
conv3_x	28×28×128	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$
conv4_x	14×14×256	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$
conv5_x	7×7×512	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$
average pool	512	7×7 average pooling
batchnorm	512	
(adaptive) projection	512	
fully connected	10	512×10 full connections
softmax	10	

Adaptive Feature Projections: Results on ResNet18



- We measure top-1 accuracy and mean average precision(mAP)
- Baselines outperform original ResNet18.
- Our methods outperform baselines and original ResNet18.
- Experiments on DenseNet121 and GoogLeNet have the same trend.
- Adaptive L∞ ball and sphere projections fail to work.

Adaptive Feature Projections: Summary

- Our methods outperform baselines consistently and significantly.
- The runtime and memory increases are negligible.

Model	Runtime (ms)	
	baseline	ours
ResNet-18	1.110	-
ResNet-18 L_1 Sphere	1.170	**1.166**
ResNet-18 L_2 Sphere	**1.634**	1.656
ResNet-18 L_1 Ball	1.164	**1.156**
ResNet-18 L_2 Ball	**1.151**	1.174

- In future, expriments on larger dataset like ImageNet.

Conclusion & Future Work

- Our adaptive methods outperform state-of-the-arts in most cases.
- The runtime and memory increases are small.

- Migrate from global pooling to local pooling.
- Parametric DDN in general:
  eg. parametric robust batchnorm for adversarial attack.

source code: https://github.com/WenboDu1228/ddn_pooling_and_projections

Thanks !

Reference

- Gould, S.; Hartley, R.;and Campbell, D., 2020. Deep declarative networks: A new hope. In *CVPR*.
- Barron, J. T., 2019. A general and adaptive robust loss function. In CVPR.
- He, K.; Zhang, X.; Ren, S.;and Sun, J., 2016. Deep residual learning for image recognition. In CVPR.
- Qi, C. R.; Su, H.; Mo, K.;andGuibas, L. J., 2016. PointNet: Deep learning on point sets for 3d classification and segmentation. In CVPR.
- Zhou, B.; Aditya Khosla, A. L.; Oliva, A.;and Torralba, A., 2016a. Learning deep features for discriminative localization. In CVPR.
- Lin, M.; Chen, Q.;and Yan, S., 2014. Network in network. In ICLR.