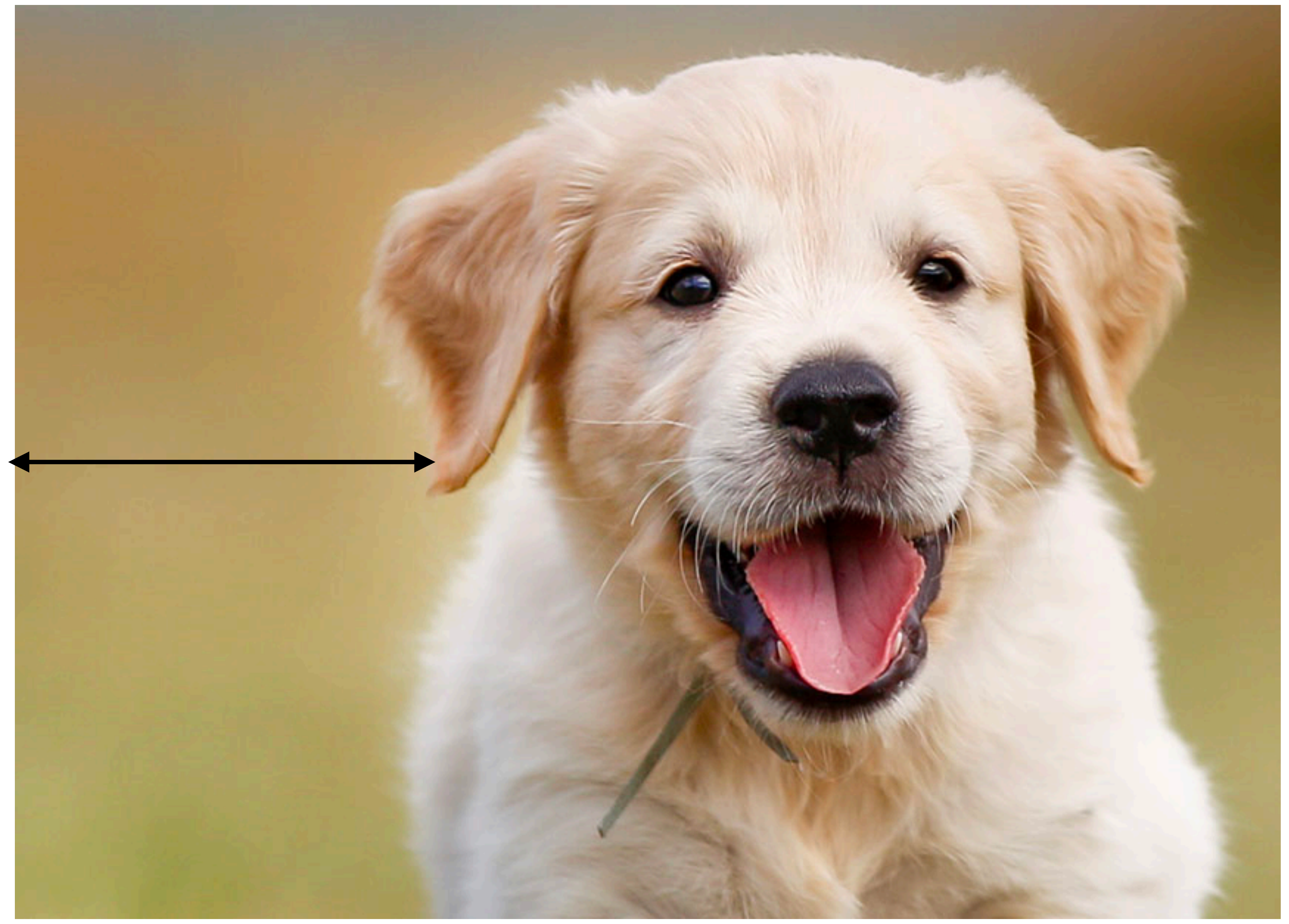
Learning to Convolve: A Generalized Weight-Tying Approach

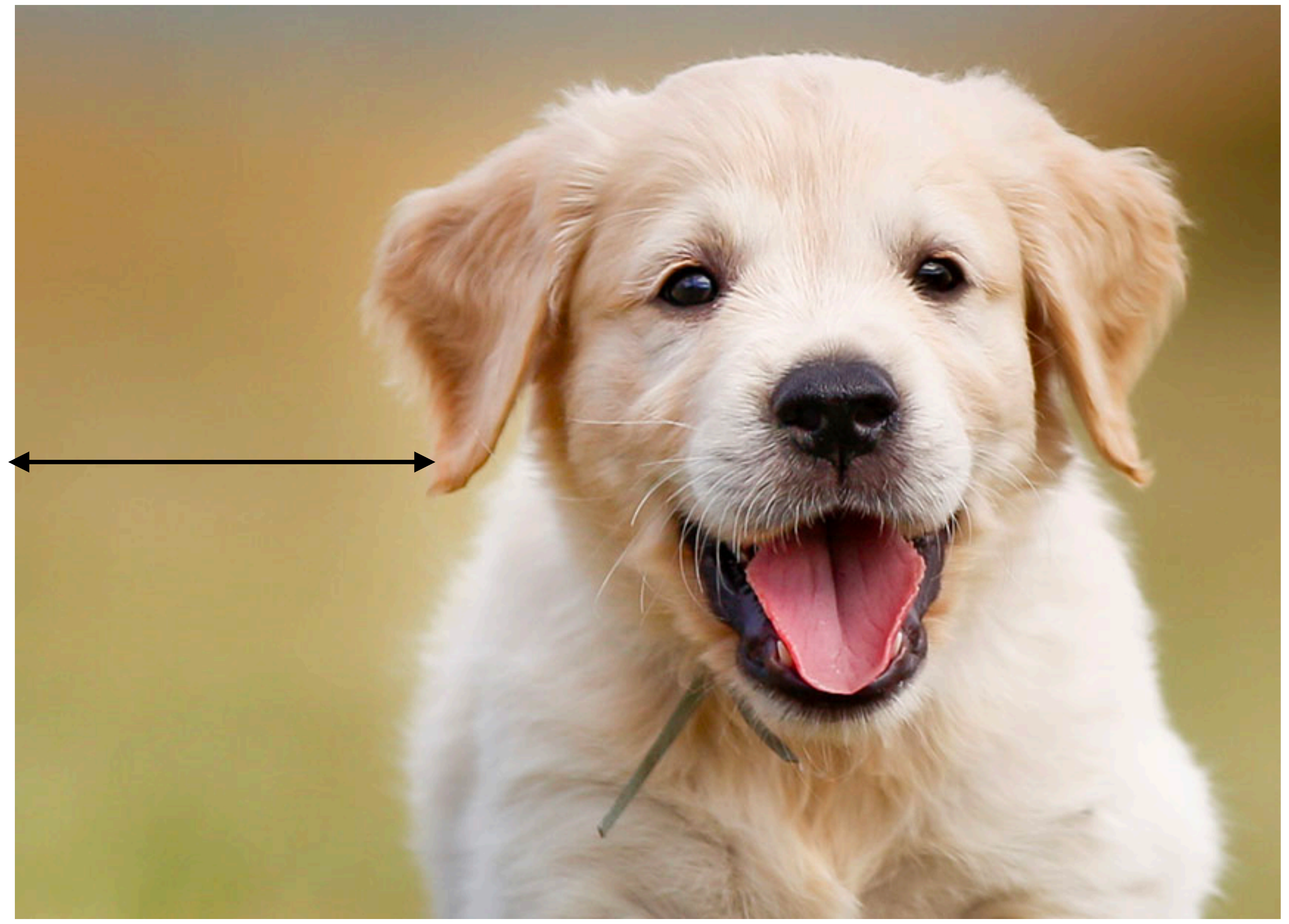
Nichita Diaconu* & Daniel Worrall*
Philips Lab ⊂ AMLAB, University of Amsterdam

ICML 2019

Symmetry

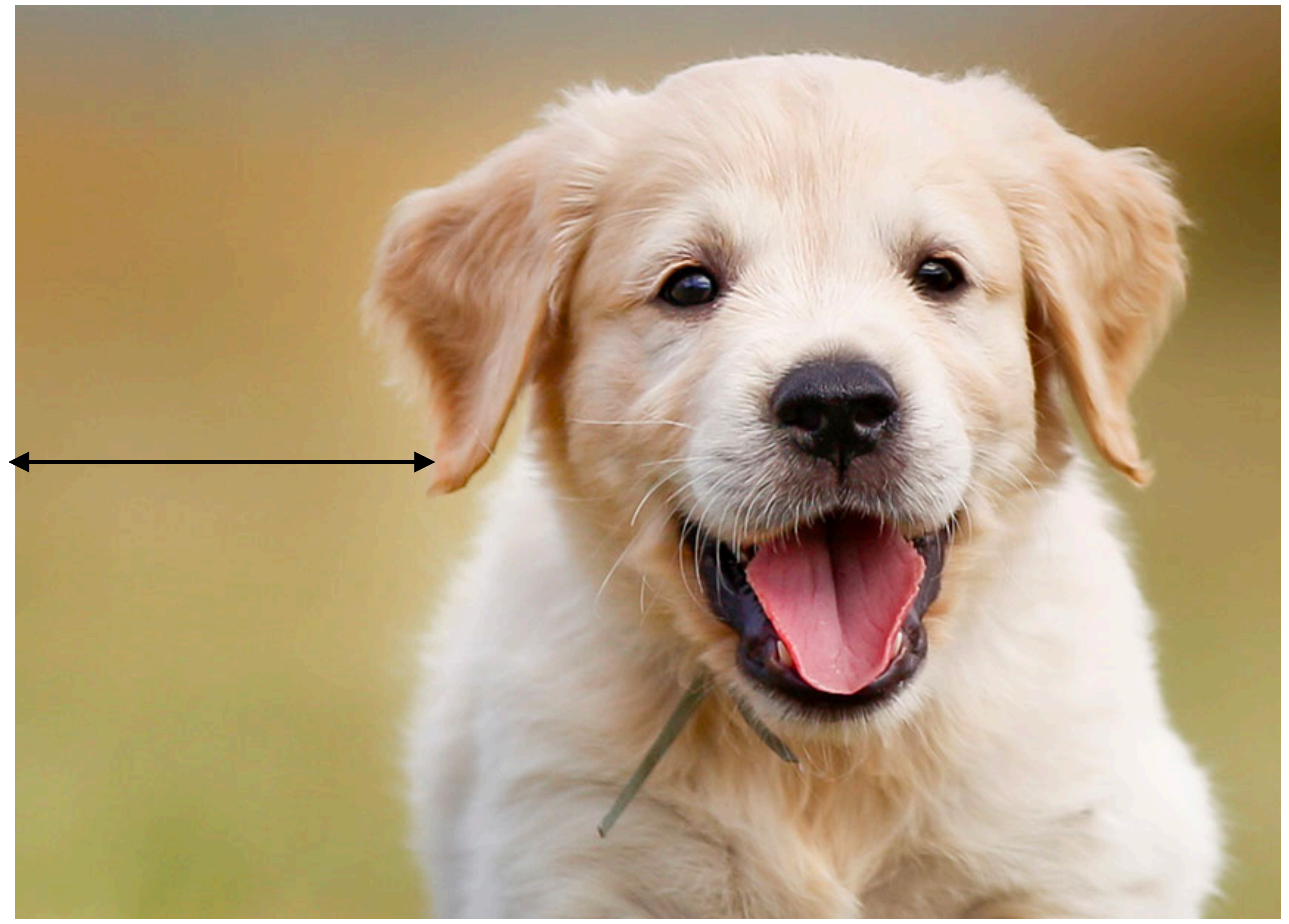


Symmetry



≠

Symmetry



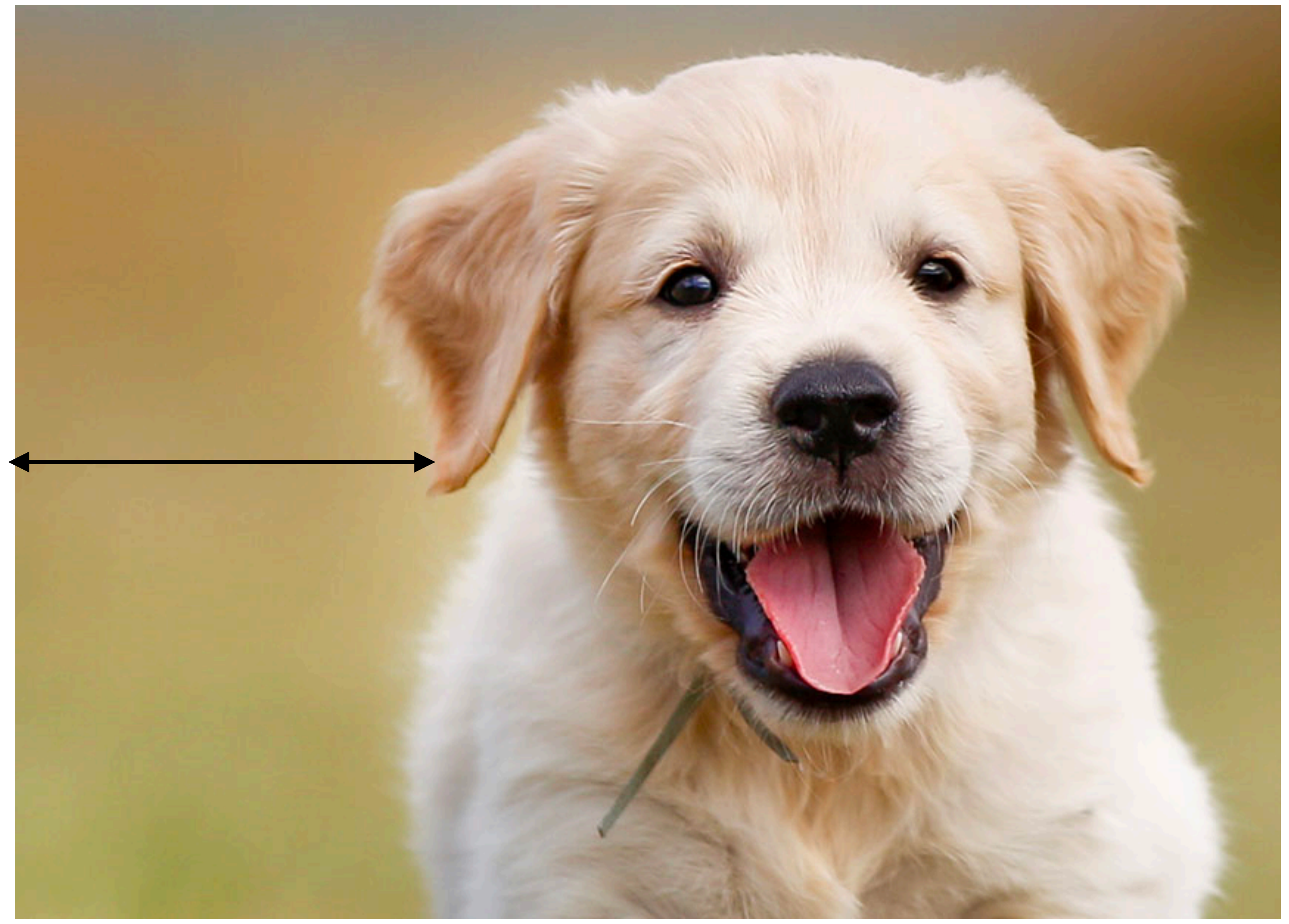
≠

Dog

Dog

Symmetry

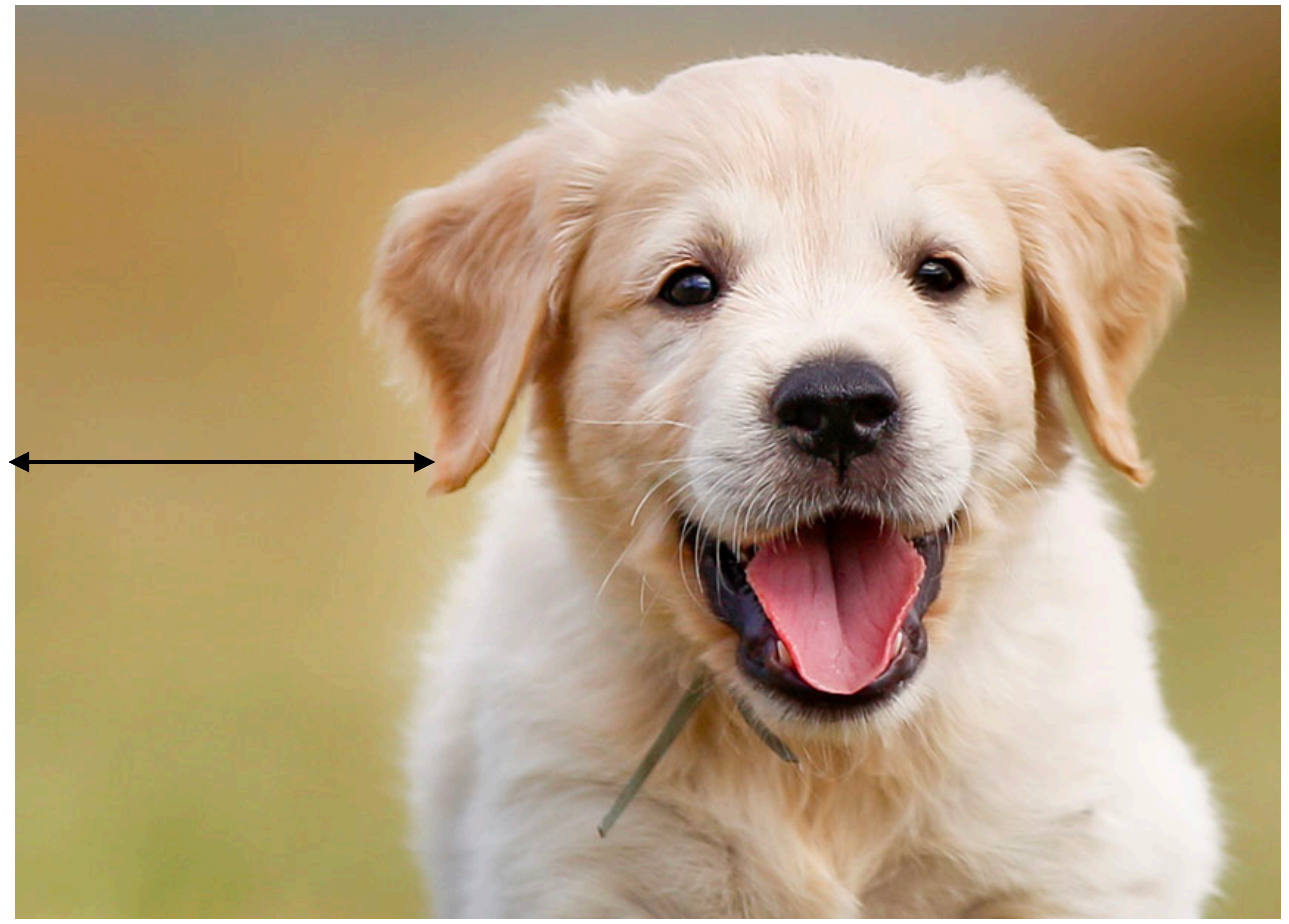
In



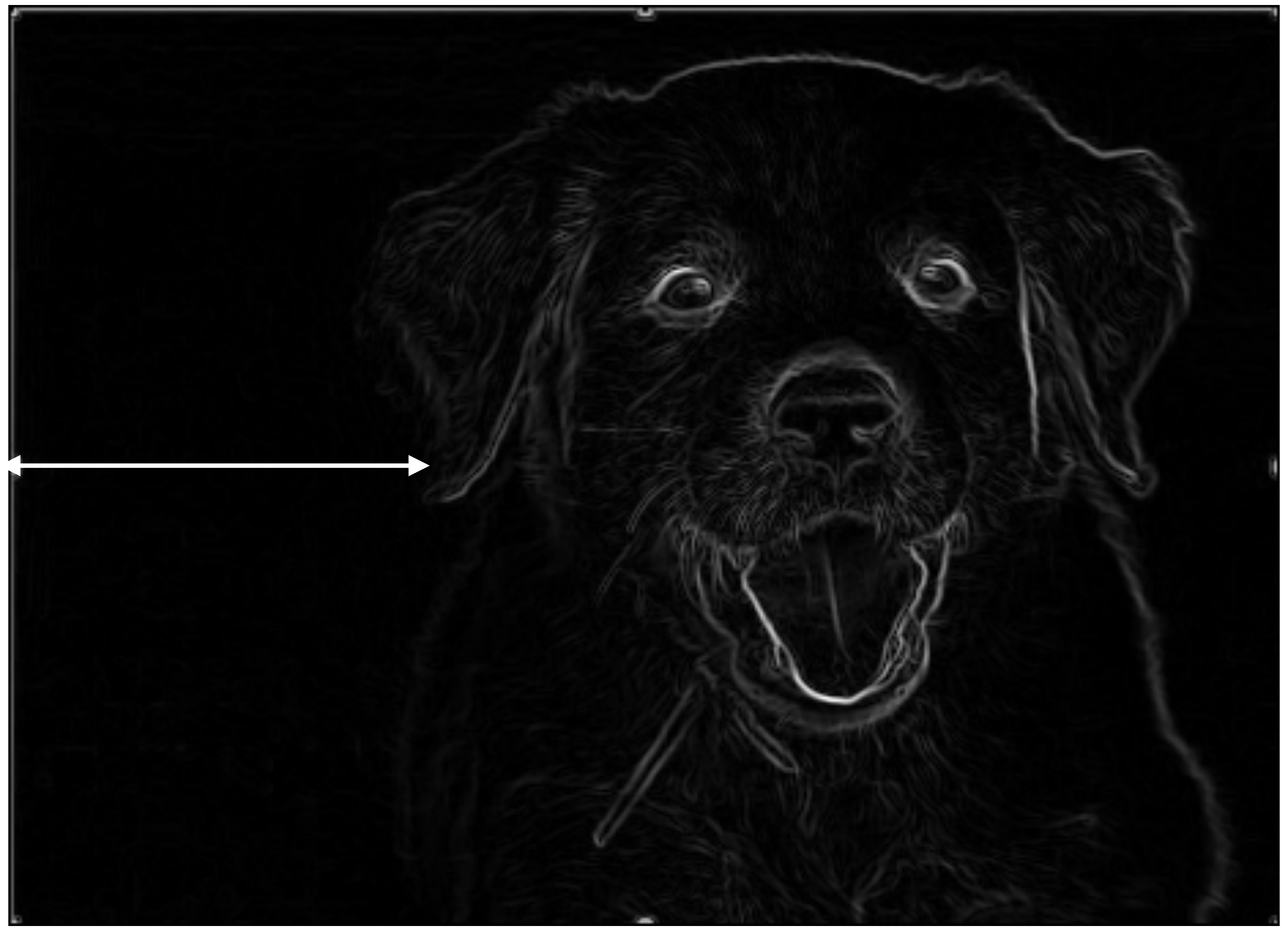
Out

Symmetry

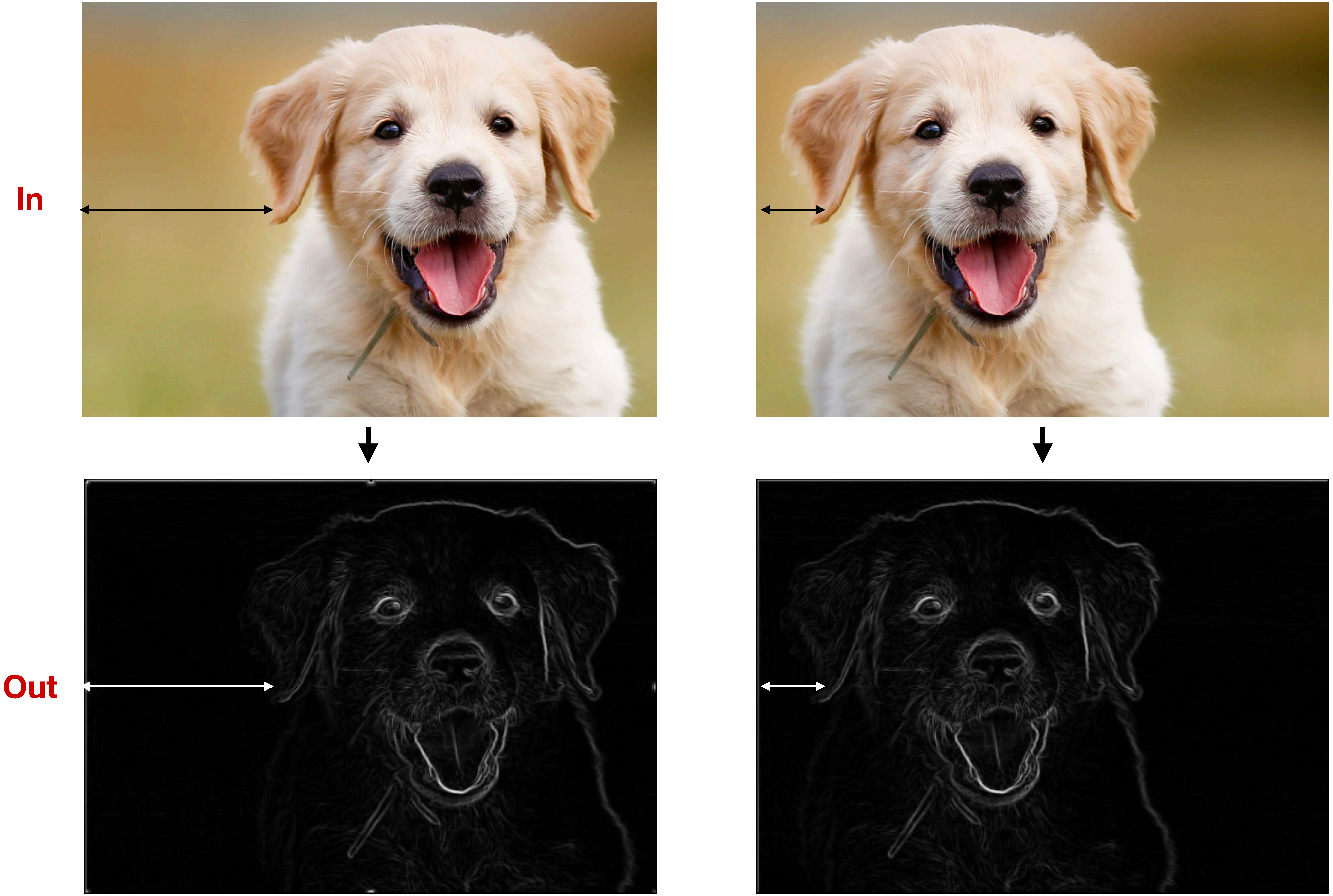
In



Out



Symmetry



Equivariance & Convolution

Standard convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x)\psi(x - g)$$

e.g. LeCun et al. (1998)

Equivariance & Convolution

Standard convolution

Input

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \psi(x - g)$$

e.g. LeCun et al. (1998)

Equivariance & Convolution

Standard convolution

Input / Filter

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x)\psi(x-g)$$

e.g. LeCun et al. (1998)

Equivariance & Convolution

Standard convolution

Input — Filter

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x)\psi(x-g)$$

e.g. LeCun et al. (1998)

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x)\mathcal{L}_g[\psi](x)$$

e.g. Cohen & Welling (2015)

Equivariance & Convolution

Standard convolution

Input Filter

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x)\psi(x-g)$$

e.g. LeCun et al. (1998)

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x)\mathcal{L}_g[\psi](x)$$

g-transformed filter

e.g. Cohen & Welling (2015)

Group Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

g-transformed filter

Group Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

g-transformed filter

$\mathcal{L}_{0^\circ}[\psi]$ $\mathcal{L}_{45^\circ}[\psi]$ $\cdots$ $\cdots$ $\mathcal{L}_{315^\circ}[\psi]$

Nearest-neighbor

Group Convolutions

Group convolution

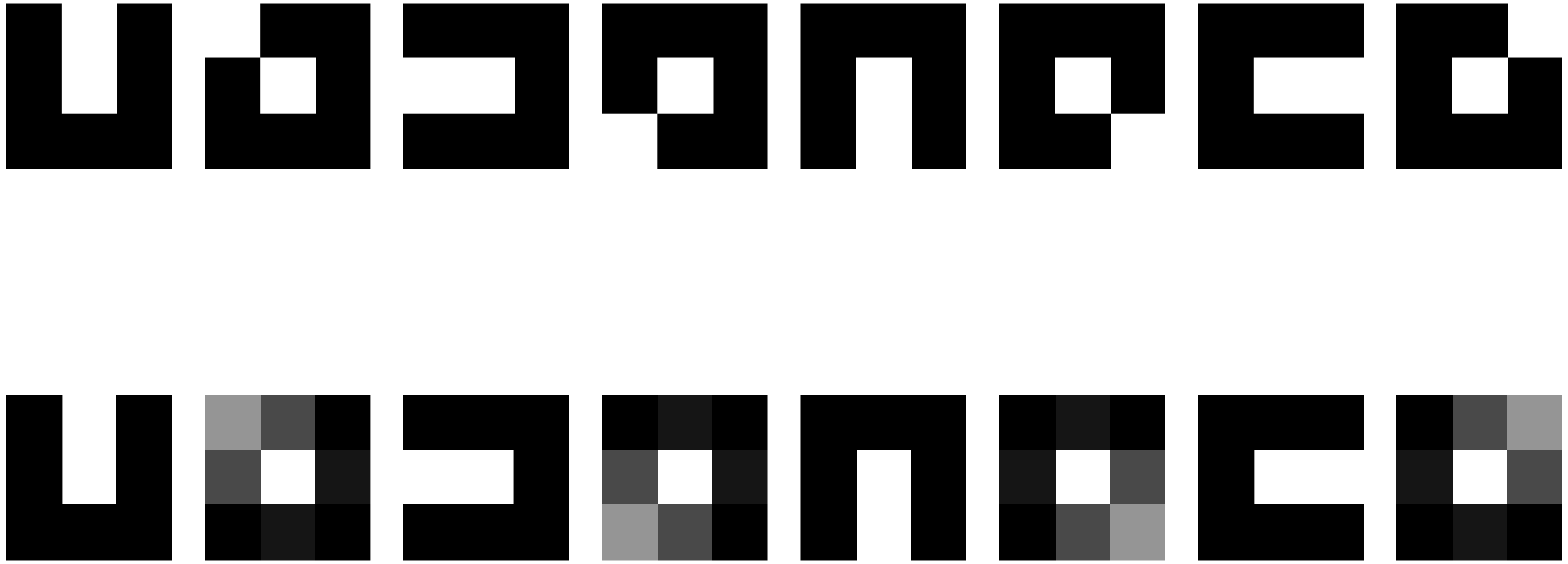
$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

g-transformed filter

$\mathcal{L}_{0^\circ}[\psi]$ $\mathcal{L}_{45^\circ}[\psi]$ $\cdots$ $\cdots$ $\mathcal{L}_{315^\circ}[\psi]$

Nearest-neighbor

Bilinear



Unitary Group Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

Unitary Group Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

Unitarity

$$\sum_{x \in \mathcal{X}} \mathcal{L}_g[f](x) \mathcal{L}_g[\psi](x) = \sum_{x \in \mathcal{X}} f(x) \psi(x)$$

Unitary Group Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

Unitarity

$$\sum_{x \in \mathcal{X}} \mathcal{L}_g[f](x) \mathcal{L}_g[\psi](x) = \sum_{x \in \mathcal{X}} f(x)\psi(x)$$

Learning Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

Learning Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

$$\psi(x) = \sum_i \hat{\psi}_i \qquad e^i(x)$$

Learning Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

$$\psi(x) = \sum_i \hat{\psi}_i \quad e^i(x)$$

Coefficients

Learning Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

$$\psi(x) = \sum_i \hat{\psi}_i \quad e^i(x)$$

Coefficients **Basis**

Learning Convolutions

Group convolution

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \mathcal{L}_g[\psi](x)$$

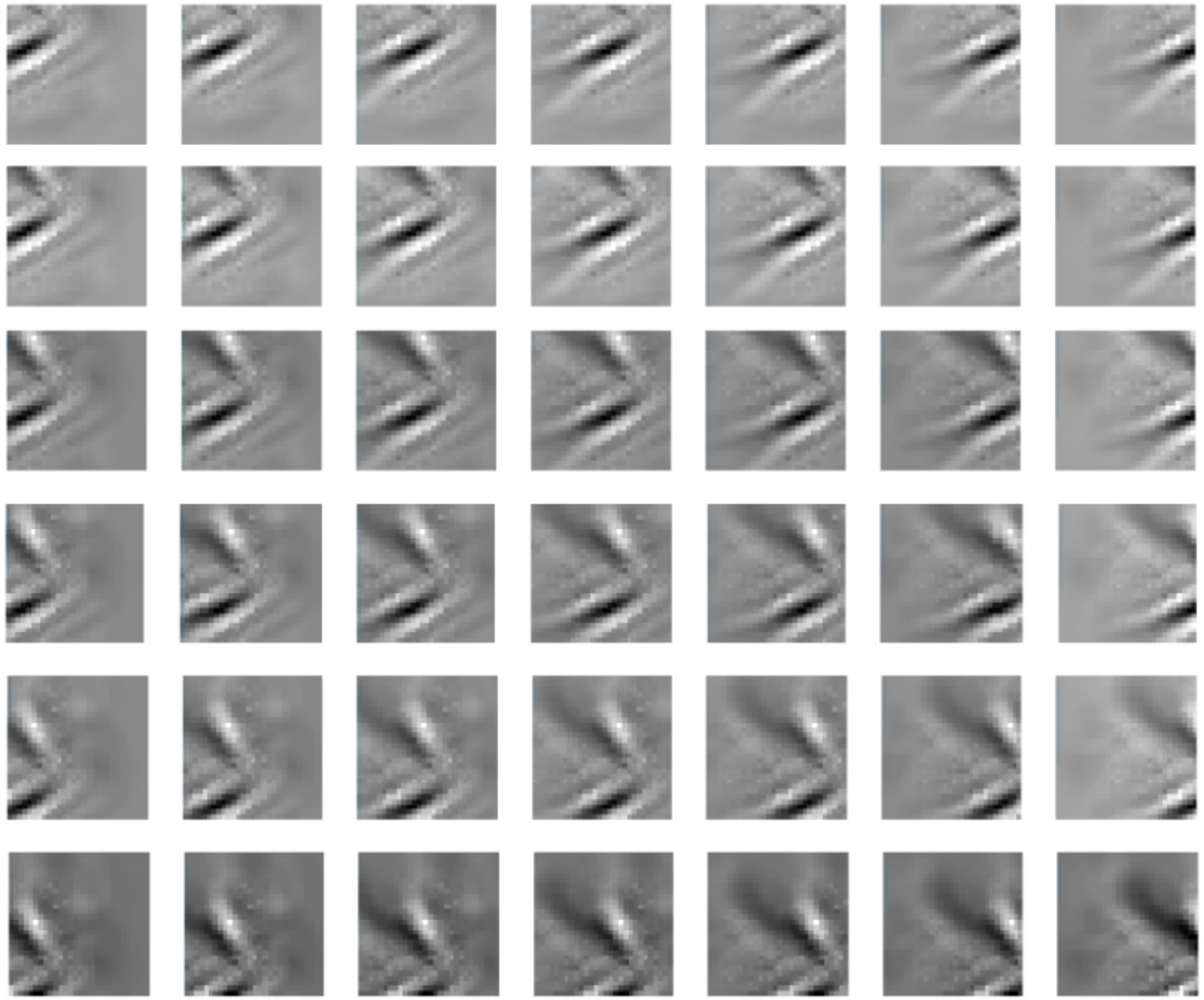
$$\mathcal{L}_g[\psi](x) = \sum_i \hat{\psi}_i \mathcal{L}_g[e^i](x)$$

Coefficients Basis

Experiments: MLP → CNN

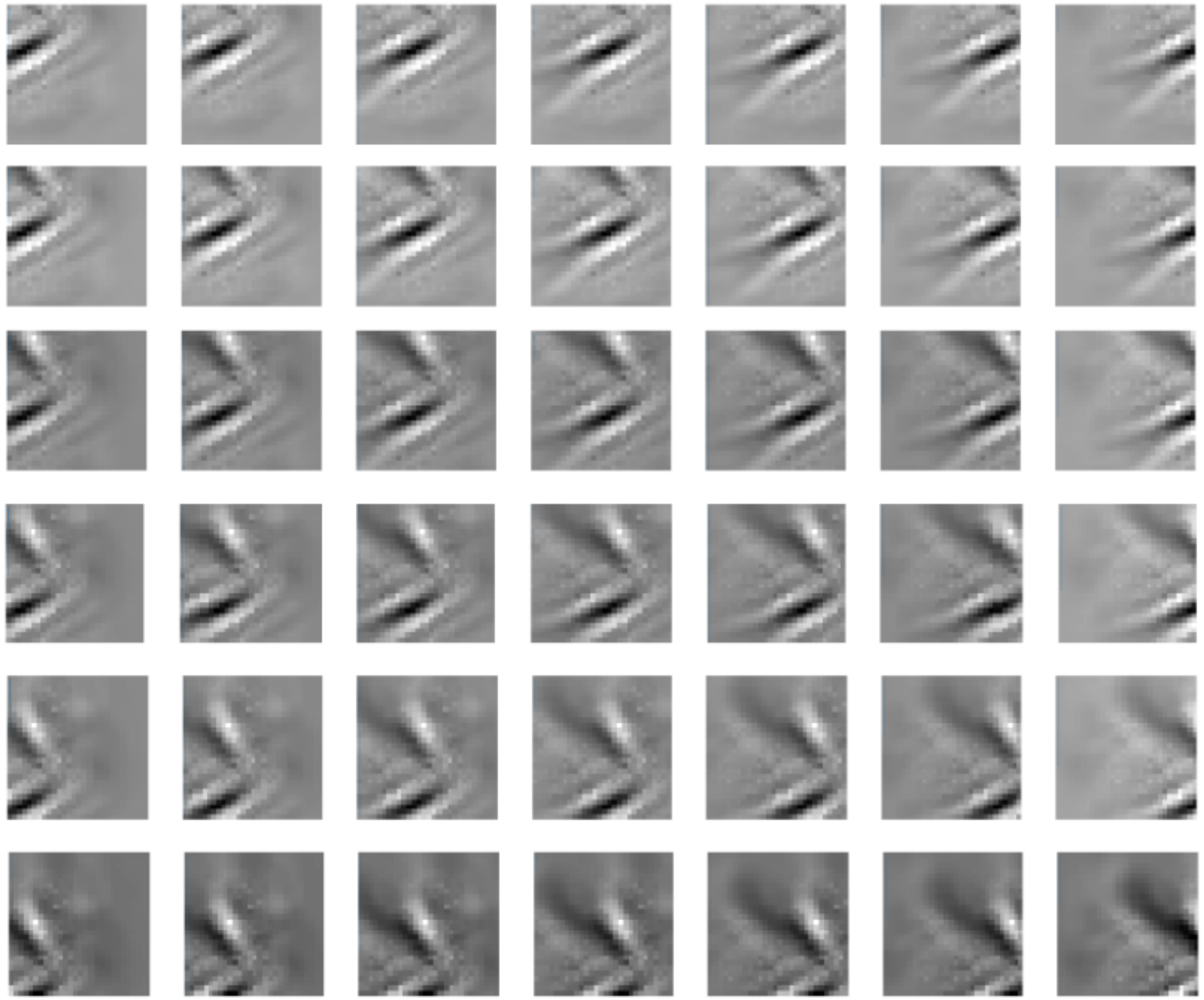
With thanks to Nichita Diaconu, Andrei Pauliuc, Daniel Maaskant, and Jens Dudink

Experiments: MLP → CNN



With thanks to Nichita Diaconu, Andrei Pauliuc, Daniel Maaskant, and Jens Dudink

Experiments: MLP → CNN



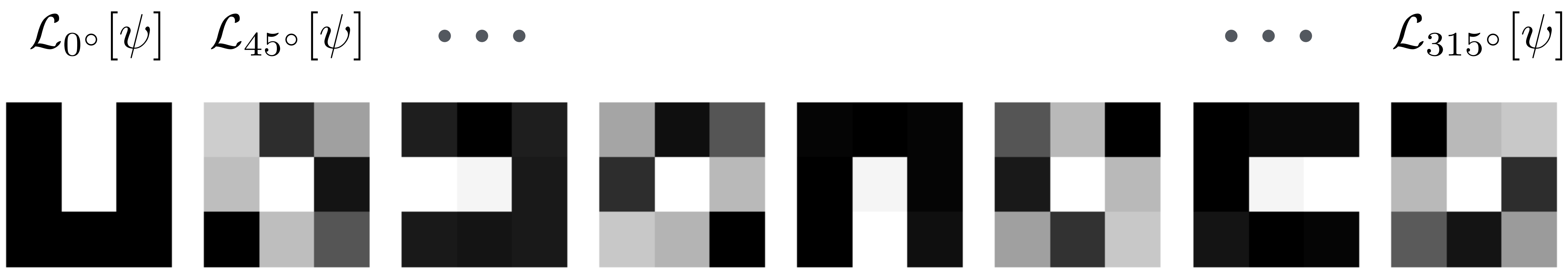
MNIST Test Error

MLP: 1.4%
Us: 0.7%
CNN: 0.5%

With thanks to Nichita Diaconu, Andrei Pauliuc, Daniel Maaskant, and Jens Dudink

Experiments: Filters

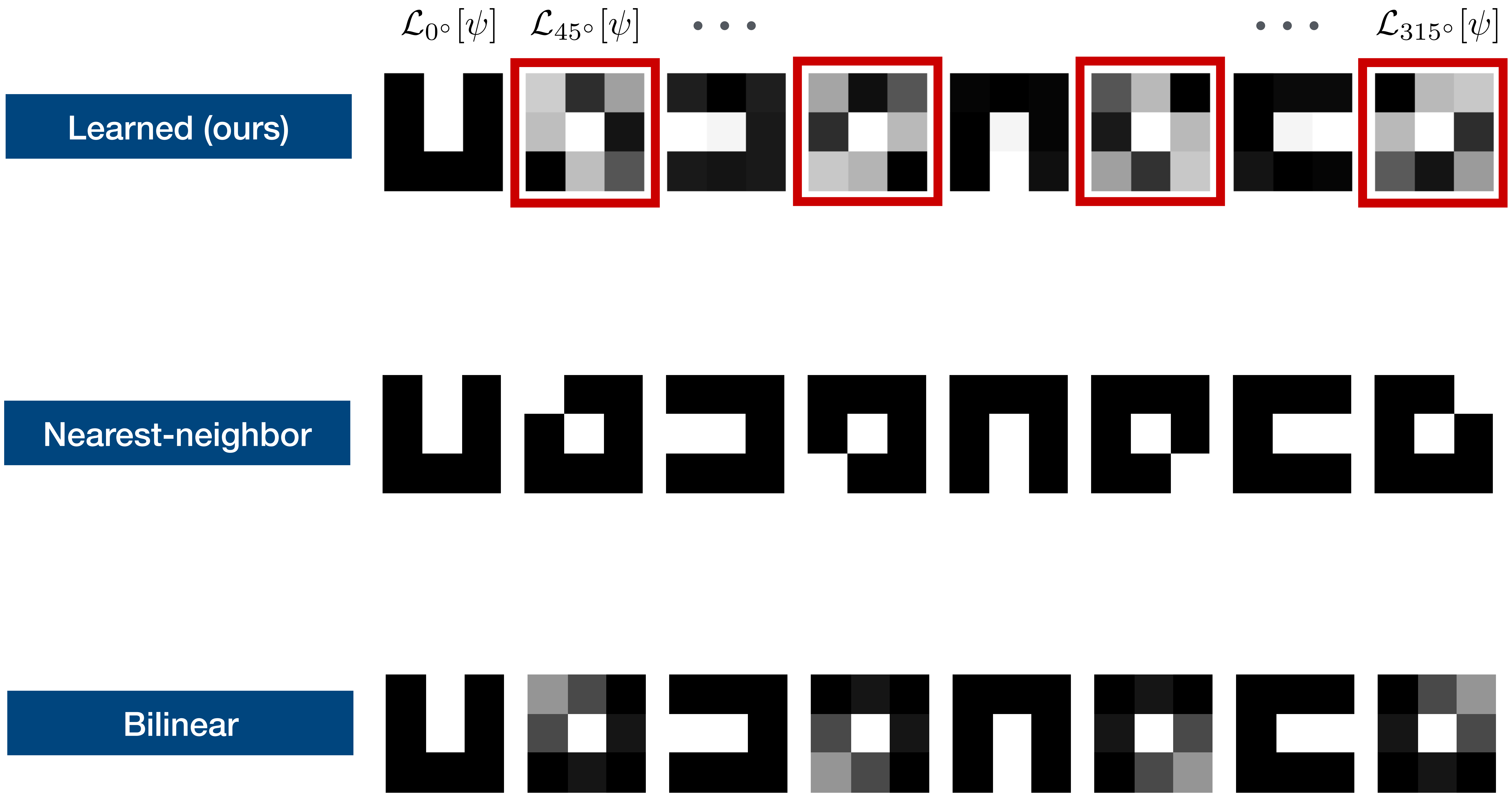
Learned (ours)



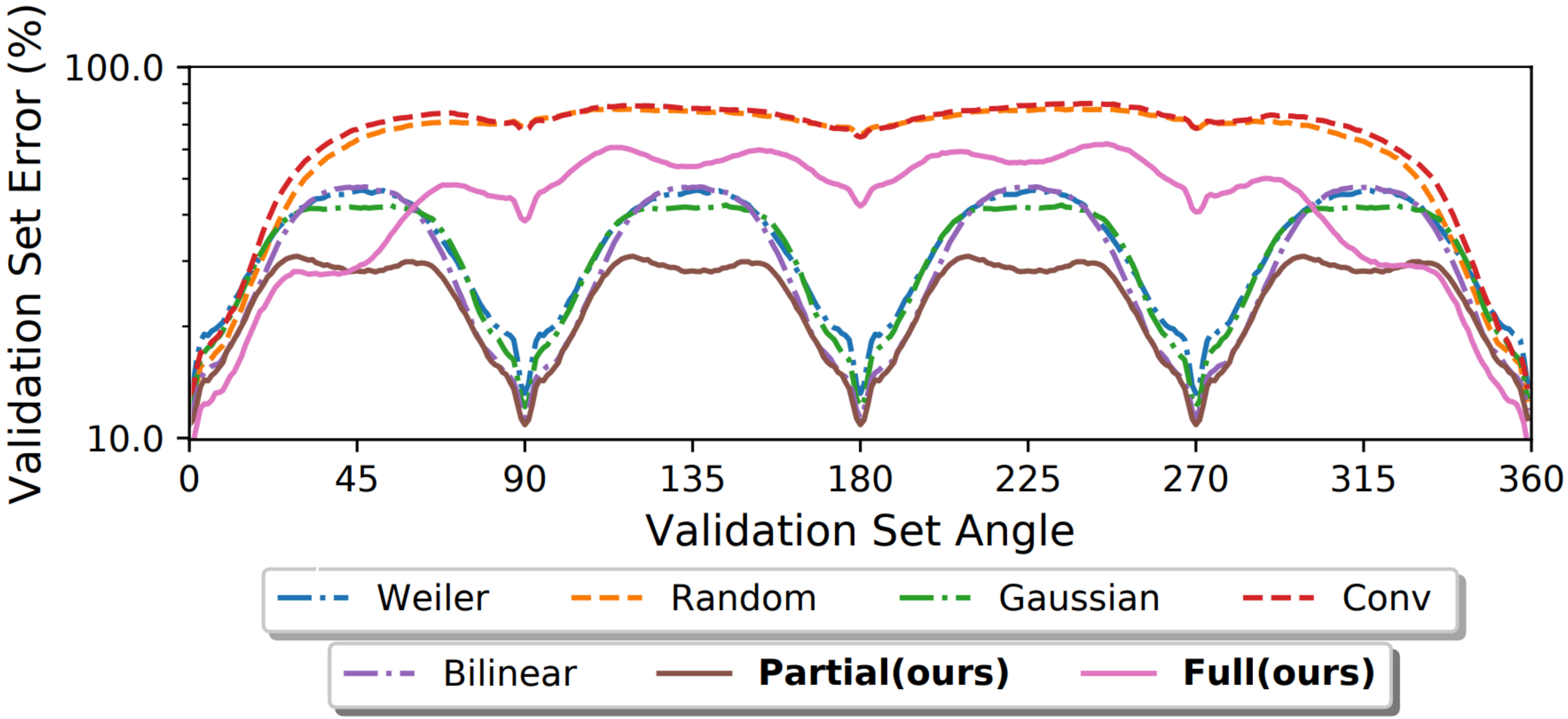
Experiments: Filters

Learned (ours)

Experiments: Filters

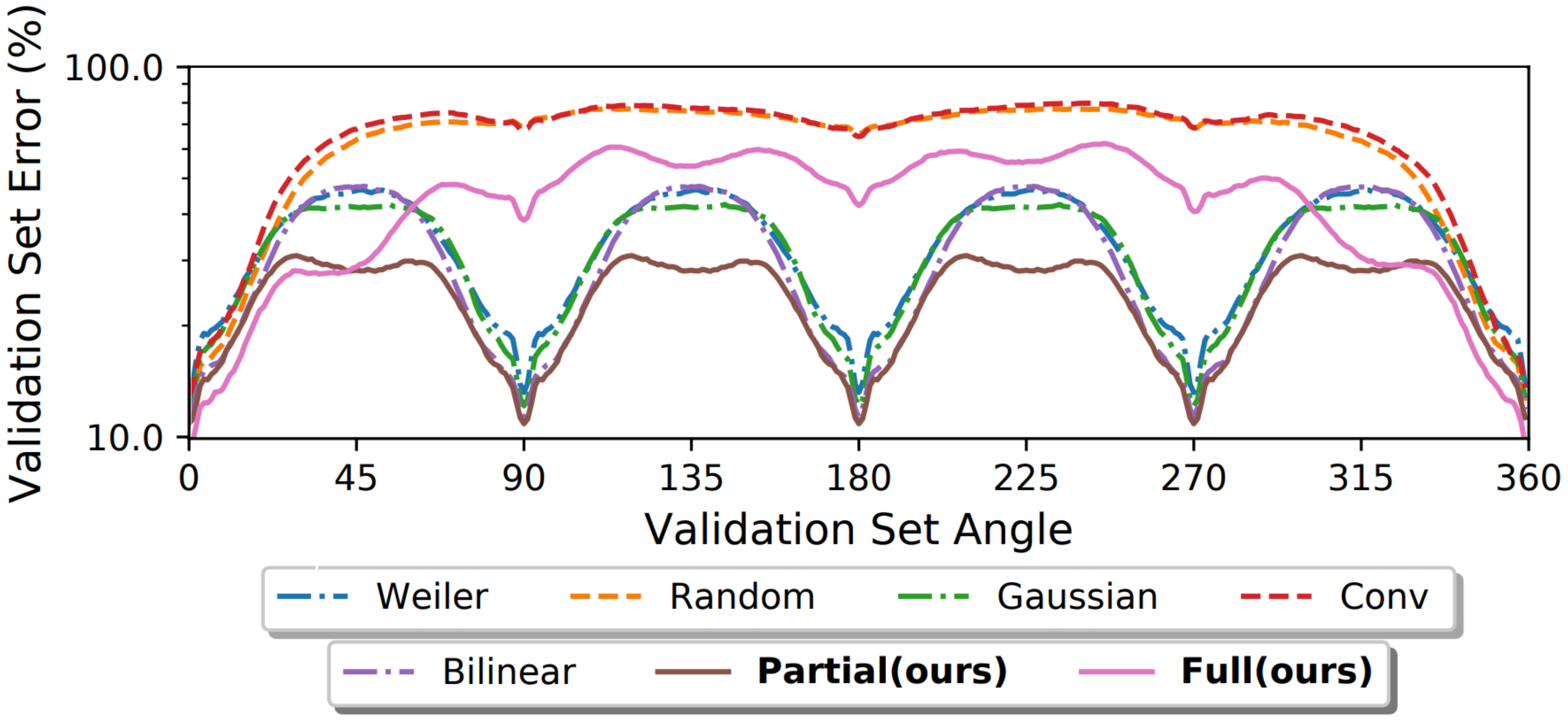


Transformation robustness



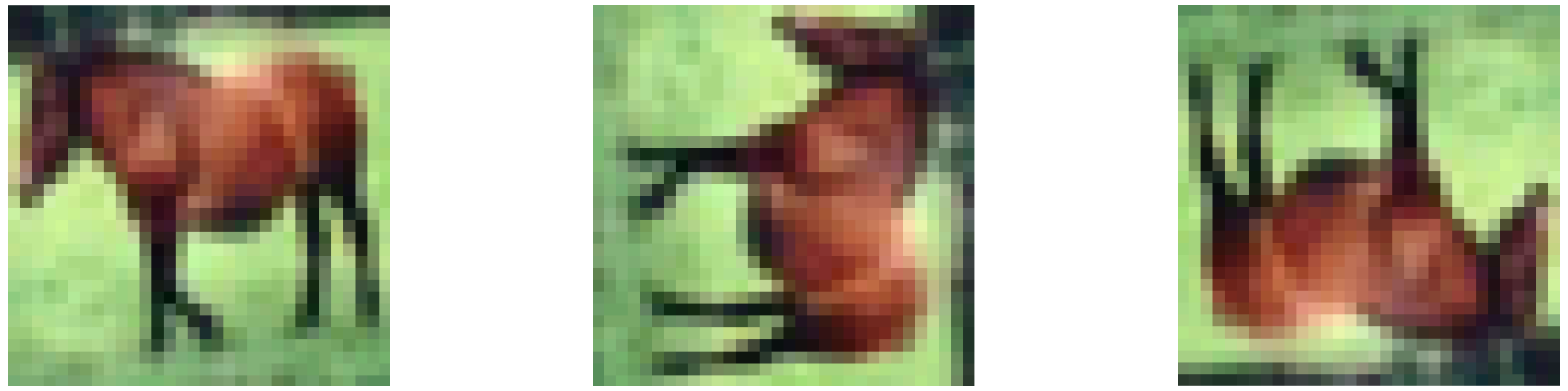
Weiler from Weiler et al. (2018)

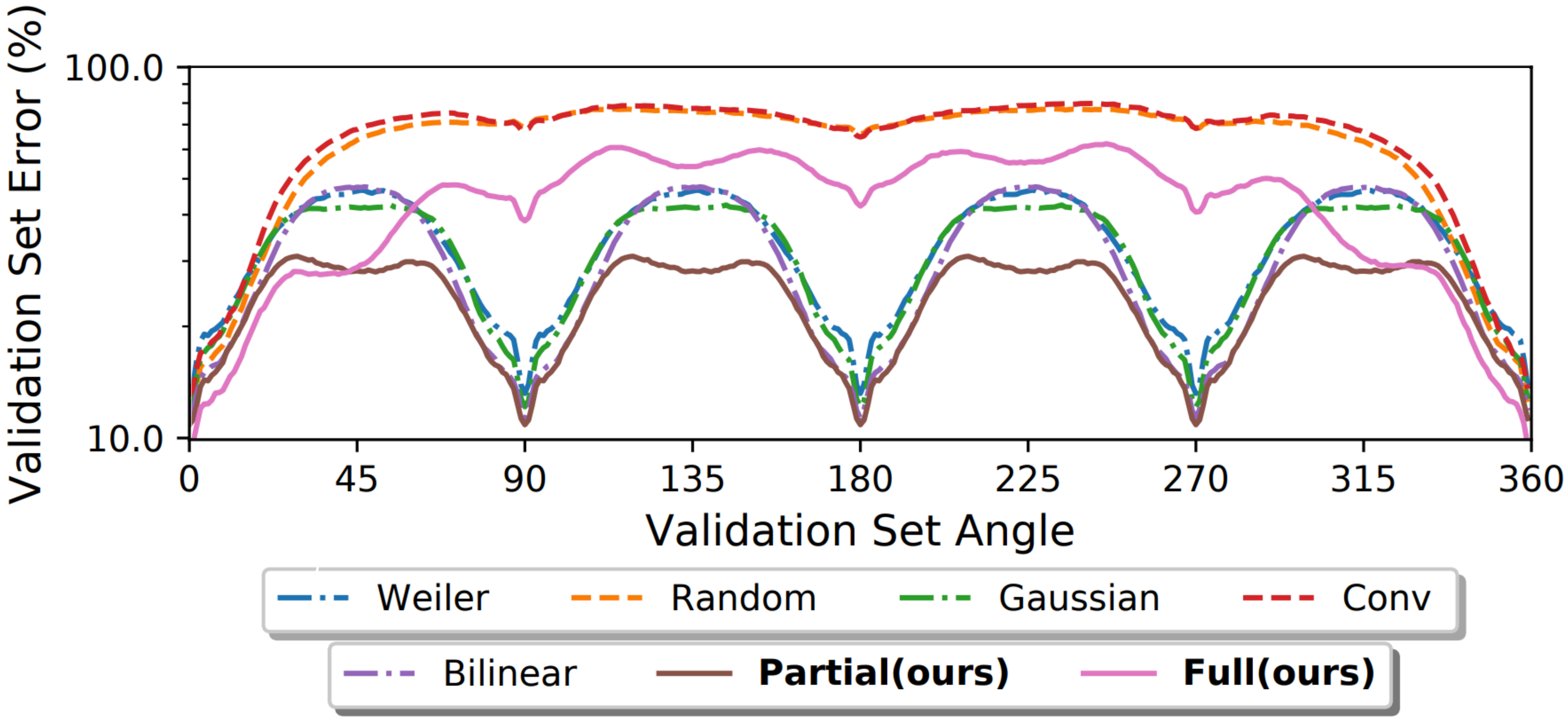
Transformation robustness



Weiler from Weiler et al. (2018)

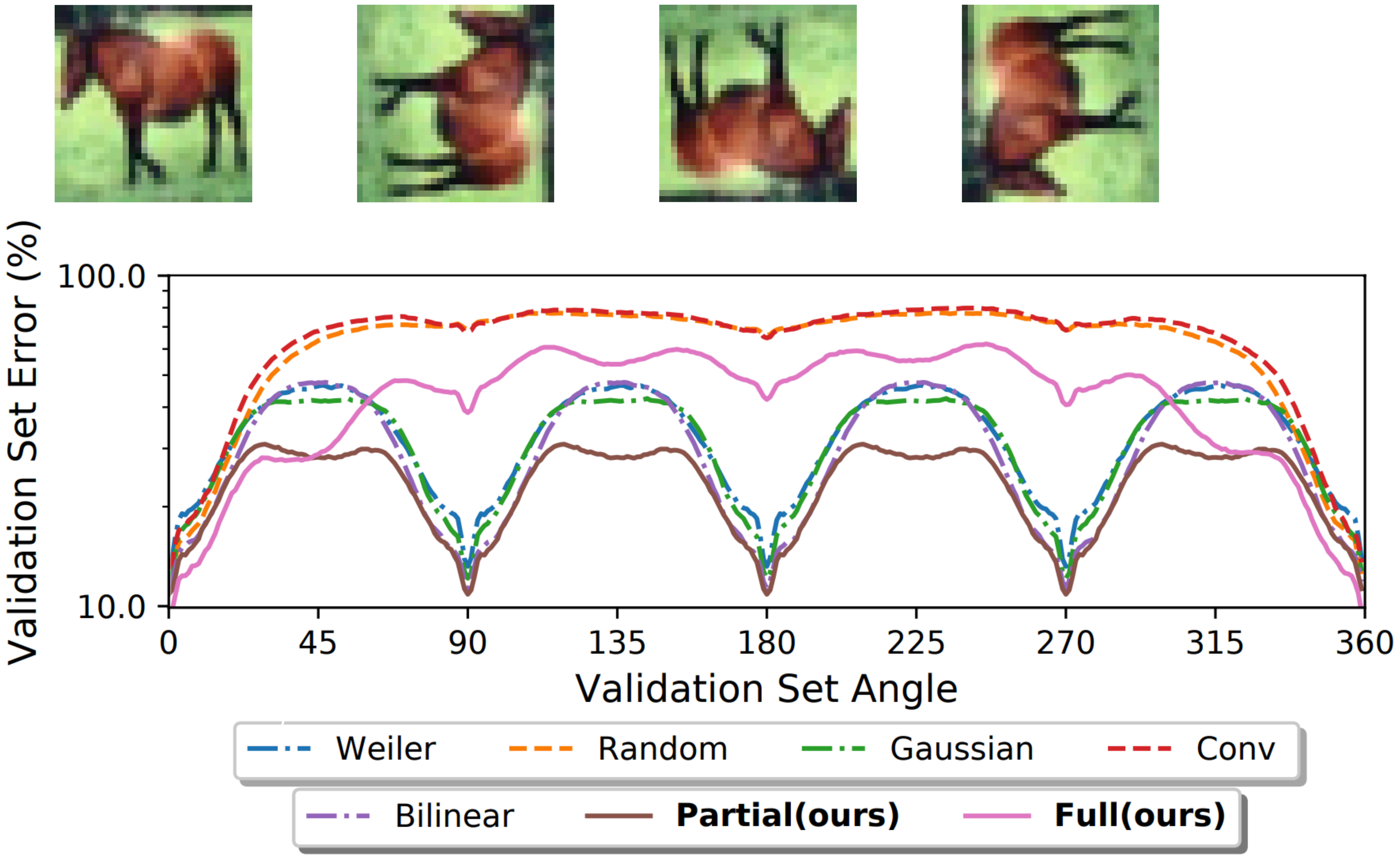
Transformation robustness





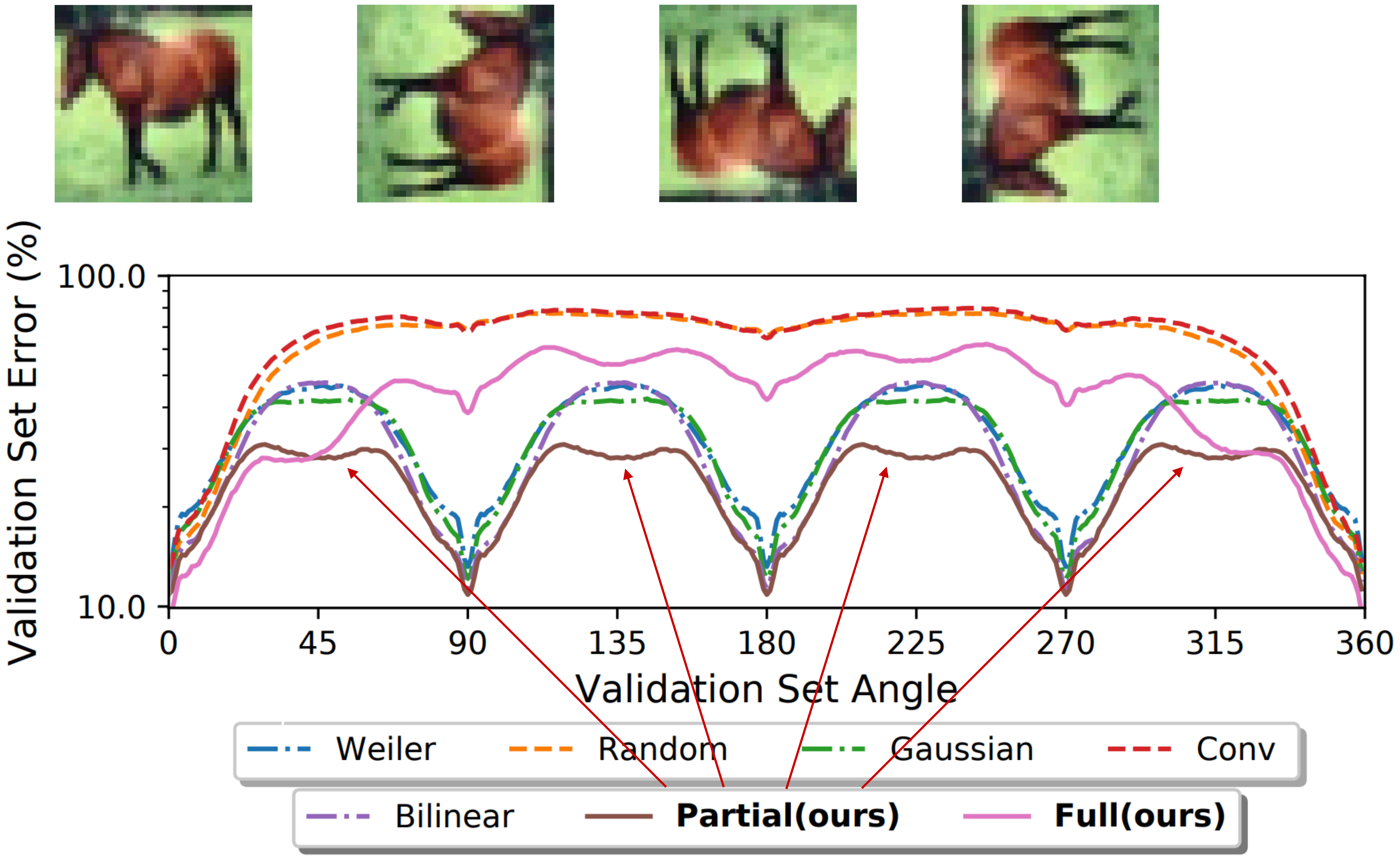
Weiler from Weiler et al. (2018)

Transformation robustness



Weiler from Weiler et al. (2018)

Transformation robustness



Weiler from Weiler et al. (2018)

Thanks

Learning to Convolve: A Generalized Weight-Tying Approach

Nichita Diaconu [1 *] Daniel Worrall [1 *]

[cs.LG] 12 May 2019

Abstract

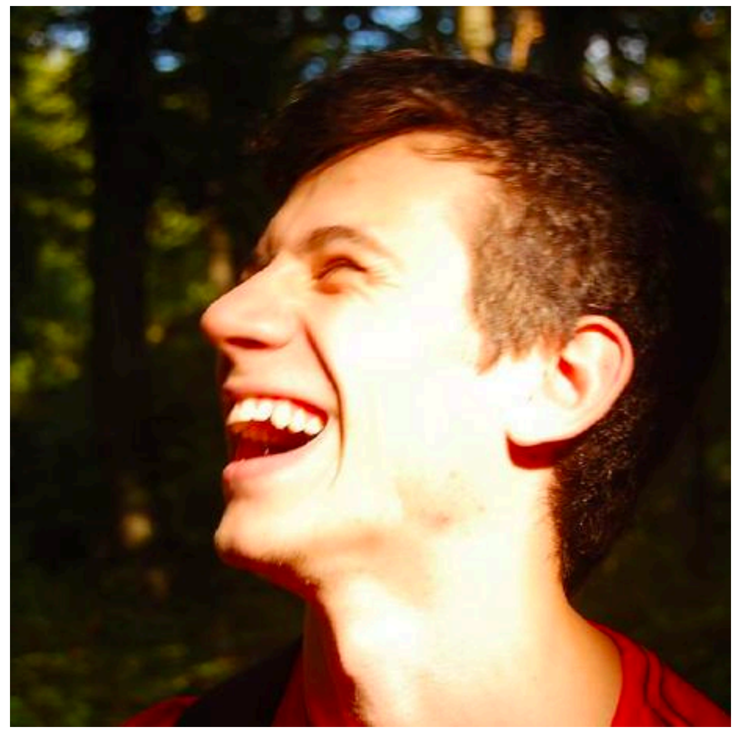
Recent work (Cohen & Welling, 2016a) has shown that generalizations of convolutions, based on group theory, provide powerful inductive biases for learning. In these generalizations, filters are not only translated but can also be rotated, flipped, etc. However, coming up with exact models of how to rotate a 3×3 filter on a square pixel-grid is difficult. In this paper, we learn how to transform filters for use in the group convolution, focussing on roto-translation. For this, we learn a filter basis and all rotated versions of that filter basis. Filters are then encoded by a set of rotation invariant coefficients. To rotate a filter, we switch the basis. We demonstrate we can produce feature maps with low sensitivity to input rotations, while achieving high performance on MNIST and CIFAR-10.

group convolutions extend standard translational convolution to the setting where the symmetry is a discrete algebraic group (explained in Section 2.2). In other words, these are convolutions over invertible transformations, so kernels are not only translated but also rotated, flipped, etc.

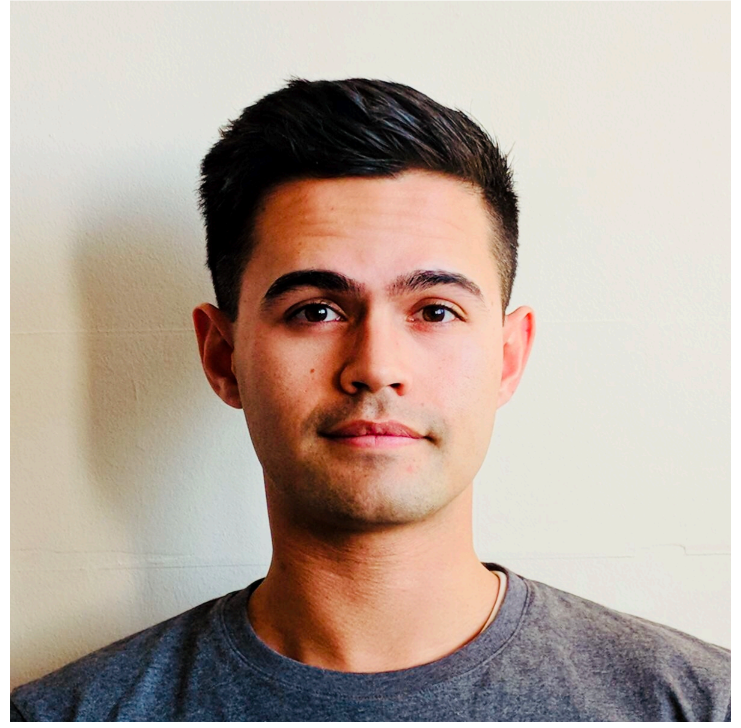
One of the key assumptions with Cohen & Welling (2016a) and associated approaches is that the set of transformations forms a group. We cannot pick an arbitrary set of transformations. For instance, in Cohen & Welling (2016a) the authors choose the group of pixelwise translations, 90° rotations, and flips, that is the set of all transformations that map the regular square-lattice into itself; and in Hoogeboom et al. (2018) the authors consider the set of all transformations that map the hexagonal lattice into itself. However, in general the set of $\frac{2\pi}{N}$ rotations for integer N and pixelwise translations does not form a group because of pixelwise discretization, yet in Bekkers et al. (2018) and Weiler et al. (2018b), the authors use these sets of transformations. Their

ICML 2019
Tues 11th Jun 2019, 18:30 - 21:00
Pacific Ballroom, Poster #78

https://deworrall92.github.io/



Nichita Diaconu



Daniel Worrall