Improving Shift Invariance in Convolutional Neural Networks with Translation Invariant Polyphase Sampling: *Supplementary Material*

Sourajit Saha
University of Maryland, Baltimore County
ssaha2@umbc.edu

Tejas Gokhale
University of Maryland, Baltimore County
gokhale@umbc.edu

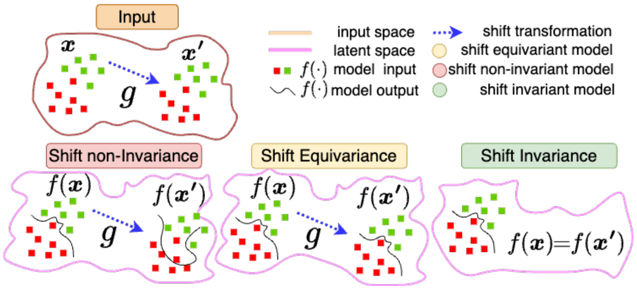

Figure 1. An illustration of shift equivariance, non-invariance, and invariance. Invariant models map shifted, non-shifted inputs to identical outputs, while equivariant models mirror the input shift in outputs.

1. Appendix

In this appendix, we define standard and circular shifts of images with examples and distinguish between shift invariance, equivariance, and non-invariance. We also discuss how polyphase decomposition operation in TIPS compares to previous works on signal propagation within CNNs for visual recognition. We further discuss computational analysis and experimental setup for MSB - shift invariance correlation study, image classification benchmarks, semantic segmentation benchmarks. Finally, we illustrate the computational overhead in TIPS and discuss how it compares to existing pooling operators.

1.1. Shift Equivariance, Invariance, and non-Invariance

Figure 1 depicts three scenarios where an input x undergoes a transformation g before being fed into a model f to generate a prediction $\hat{y} = f(g(x)) = g'(f(x))$: shift equivariance, shift non-invariance, and shift invariance. If $g' = g$, then f is g-equivariant and if $g' = I$ then f is g-invariant. Shift-invariance is desirable for image classification to ensure that categorical outputs are invariant to pixel shift, and shift-equivariance is desirable for semantic segmentation and object detection to ensure that pixel-shift in the image results in equivalent shift in corresponding segmentation masks and bounding boxes.

1.2. Standard and Circular Shifts of Images

There are two types of pixel levels shifts that can be performed on images: standard shift and circular shift. Given an image of height h and width w, we can perform either type of shifts by an amount (x, y) where $x \in \{0, .., h\}$, $y \in \{0, .., w\}$. Standard shift is the process of shifting images to a (x, y) direction which renders blank pixels at shifted positions. Circular shift also shifts images in the (x, y) direction, except the shifted pixels that move beyond the image boundary, are wrapped about the opposite ends of the image to fill in the empty pixels. Therefore, circular shift is a lossless transformation while standard shift is not. Figure 2 and 3 show examples of standard and circular shift (by varying amounts) applied to an image taken from ImageNet test set and depict how standard shift renders blank pixels while circular shift do not.

1.3. Comparison of Polyphase Decomposition operation in the TIPS layer with previous work in signal propagation within CNNs for image recognition

Within TIPS layers, we use polyphase decomposition which is comparable to dilated convolution (13) and dilated attention (6) where the stride and dilation rates are identical (Fig 2 in manuscript). Usage of strided convolution in the above convolution operations can also be used for spatial downsampling, however strided convolutions are still shift invariant (14). The slicing operation in polyphase decomposition is also identical to that of parallel grid pooling (10), focus layer in YOLOv5 (4). However, we learn to sample from these polyphase decompositions in the channel dimension while (4; 10) stack these decompositions in the channel space and then uses group convolution to downsample across the channel dimension.

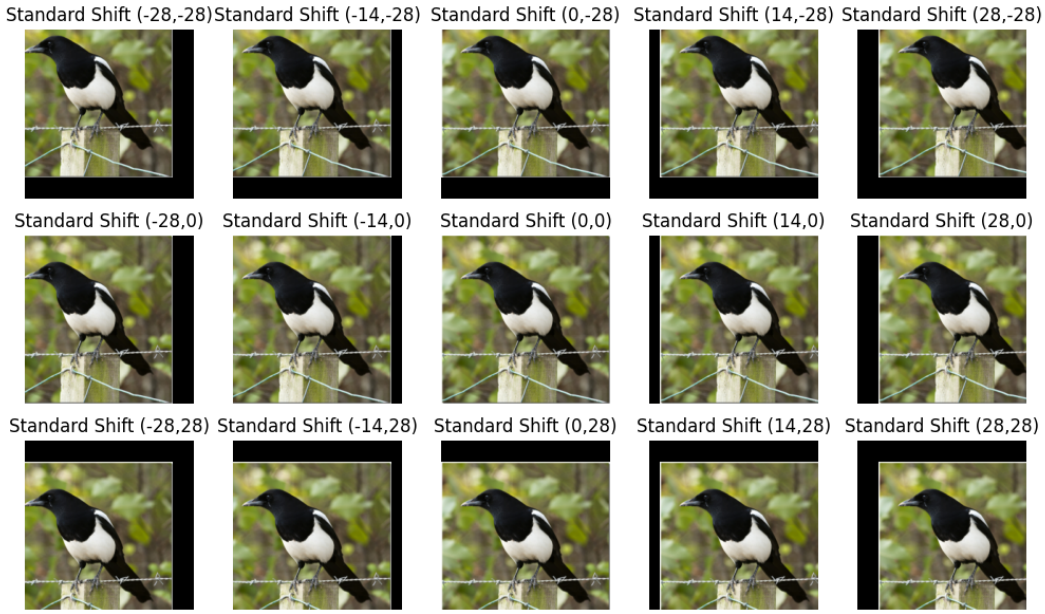

Figure 2. Standard shift of an 224 × 224 image from ImageNet test set is shown with varying amount of shifts. Here, standard shift $(0, 0)$ denotes the original image with no shifts. It is also observed that, as the amount of standard shift increases, there occurs more information (pixel) loss.

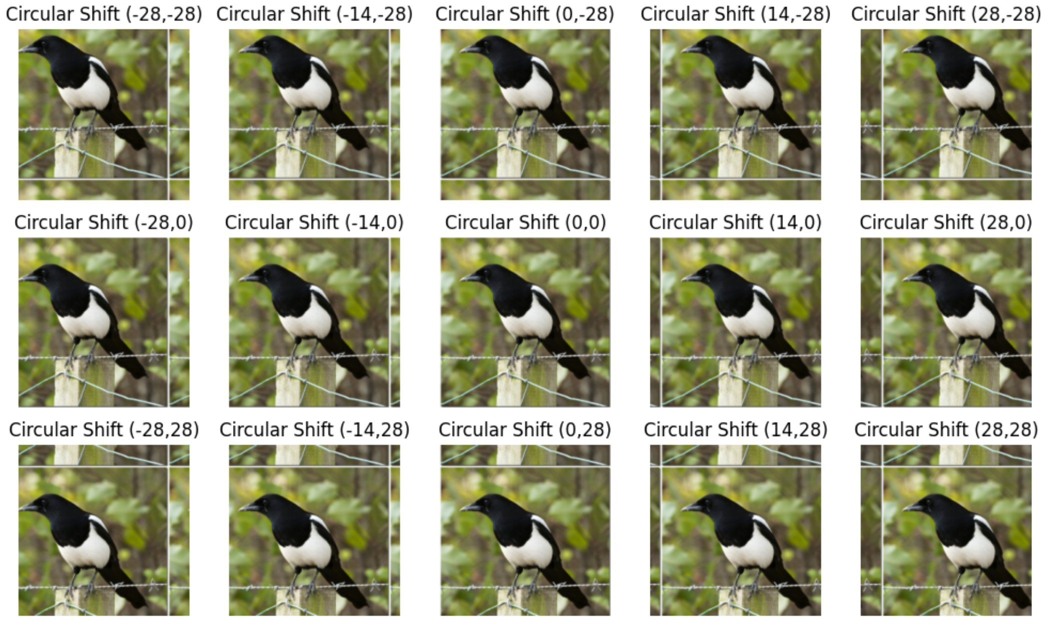

Figure 3. Circular shift of an 224 × 224 image from ImageNet test set is shown with varying amount of shifts. Here, circular shift $(0, 0)$ denotes the original image with no shifts.

1.4. Comparison of TIPS with existing Polyphase Sampling Pooling

While TIPS, APS, and LPS use polyphase decomposition for spatial downsampling, they differ in how the pooled features are sampled from the decomposed polyphase components.

- *APS* simply samples the polyphase component that contains the maximum energy using ℓ_p norm.
- *LPS* learns to sample from these polyphase components

		Unshifted	Standard Shift		Circular Shift	
	Method	**Acc. ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**
CNN (ResNet-50)	MaxPool	92.96$_{\pm0.08}$	82.13$_{\pm0.57}$	76.18$_{\pm0.07}$	83.61$_{\pm0.12}$	77.72$_{\pm0.05}$
	APS	94.68$_{\pm0.11}$	91.34$_{\pm0.04}$	86.48$_{\pm0.13}$	**100.00**$_{\pm0.00}$	94.68$_{\pm0.11}$
	LPS	94.71$_{\pm0.02}$	92.41$_{\pm0.03}$	87.52$_{\pm0.11}$	99.48$_{\pm0.11}$	94.22$_{\pm0.05}$
	TIPS	**95.63**$_{\pm0.15}$	**95.02**$_{\pm0.09}$	**90.87**$_{\pm1.08}$	**100.00**$_{\pm0.00}$	**95.63**$_{\pm0.15}$
	BlurPool (LPF-5)	93.77$_{\pm0.03}$	88.18$_{\pm0.17}$	82.69$_{\pm1.08}$	93.49$_{\pm0.13}$	87.67$_{\pm0.03}$
	APS (LPF-5)	94.07$_{\pm0.13}$	92.51$_{\pm0.06}$	87.03$_{\pm0.20}$	**100.00**$_{\pm0.00}$	94.07$_{\pm0.13}$
	LPS (LPF-5)	95.62$_{\pm0.07}$	94.10$_{\pm0.07}$	89.99$_{\pm0.19}$	**100.00**$_{\pm0.00}$	95.62$_{\pm0.07}$
	TIPS (LPF-5)	**96.42**$_{\pm0.16}$	**95.50**$_{\pm0.13}$	**92.08**$_{\pm0.19}$	**100.00**$_{\pm0.00}$	**96.42**$_{\pm0.16}$
ViT	ViT-B/16 (I21k)	96.88$_{\pm0.13}$	81.45$_{\pm0.04}$	78.91$_{\pm0.15}$	**78.39**$_{\pm0.12}$	75.94$_{\pm0.12}$
	ViT-L/16 (I21k)	97.00$_{\pm0.03}$	81.84$_{\pm0.11}$	79.38$_{\pm0.08}$	78.06$_{\pm0.18}$	75.72$_{\pm0.17}$
	Swin-B (I21k)	**97.49**$_{\pm0.05}$	**82.85**$_{\pm0.14}$	**80.77**$_{\pm0.09}$	78.05$_{\pm0.02}$	**76.10**$_{\pm0.08}$

(a) Food-101

		Unshifted	Standard Shift		Circular Shift	
	Method	**Acc. ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**
CNN (ResNet-50)	MaxPool	93.48$_{\pm0.15}$	85.63$_{\pm0.11}$	80.05$_{\pm0.17}$	89.38$_{\pm0.17}$	83.55$_{\pm0.12}$
	APS	94.68$_{\pm0.03}$	92.47$_{\pm0.05}$	87.55$_{\pm1.09}$	**100.00**$_{\pm0.00}$	94.68$_{\pm0.03}$
	LPS	95.31$_{\pm0.08}$	93.63$_{\pm0.17}$	89.24$_{\pm0.11}$	**100.00**$_{\pm0.00}$	95.31$_{\pm0.08}$
	TIPS	**97.18**$_{\pm0.06}$	**95.78**$_{\pm0.03}$	**93.08**$_{\pm0.16}$	**100.00**$_{\pm0.00}$	**97.18**$_{\pm0.06}$
	BlurPool (LPF-5)	92.71$_{\pm0.08}$	90.32$_{\pm0.13}$	83.74$_{\pm0.05}$	94.07$_{\pm0.13}$	87.21$_{\pm0.08}$
	APS (LPF-5)	94.71$_{\pm0.11}$	93.00$_{\pm0.08}$	88.09$_{\pm0.14}$	**100.00**$_{\pm0.00}$	94.71$_{\pm0.11}$
	LPS (LPF-5)	96.28$_{\pm0.05}$	94.33$_{\pm0.06}$	90.82$_{\pm0.09}$	**100.00**$_{\pm0.00}$	96.28$_{\pm0.05}$
	TIPS (LPF-5)	**97.62**$_{\pm0.11}$	**96.51**$_{\pm0.14}$	**94.21**$_{\pm0.14}$	**100.00**$_{\pm0.00}$	**97.62**$_{\pm0.11}$
ViT	ViT-B/16 (I21k)	99.33$_{\pm0.05}$	**88.47**$_{\pm0.04}$	**87.88**$_{\pm0.08}$	82.24$_{\pm0.03}$	81.69$_{\pm0.06}$
	ViT-L/16 (I21k)	99.59$_{\pm0.03}$	87.25$_{\pm0.09}$	86.89$_{\pm0.18}$	82.39$_{\pm0.13}$	82.05$_{\pm0.03}$
	Swin-B (I21k)	**99.68**$_{\pm0.02}$	87.06$_{\pm0.16}$	80.16$_{\pm0.07}$	**83.57**$_{\pm0.11}$	**83.30**$_{\pm0.05}$

(b) Oxford-102

Table 1. Image classification performance on Food-101 and Oxford-102 datasets averaged over five trials.

		Kvasir - U-Net					CVC-ClinicDB - U-Net				
		Unshifted	**Standard Shift**		**Circular Shift**		**Unshifted**	**Standard Shift**		**Circular Shift**	
Method	**Anti-Alias**	**mIOU ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**	**mIOU ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**
MaxPool	-	75.60	92.84	70.19	97.91	74.02	73.81	90.24	66.61	95.50	70.50
Blurpool	LPF-3	78.39	94.63	74.18	98.30	77.06	76.32	93.87	71.64	96.36	73.54
DDAC	LPF-3	79.24	95.17	75.41	98.49	78.04	77.89	92.17	71.80	97.73	76.12
APS	LPF-3	81.97	96.32	78.95	**100.00**	81.97	79.31	95.63	75.84	**100.00**	79.31
LPS	LPF-3	82.38	97.86	80.62	**100.00**	82.38	78.59	96.21	75.61	**100.00**	78.59
TIPS	LPF-3	**86.10**	**98.09**	**84.46**	**100.00**	**86.10**	**80.05**	**97.89**	**78.36**	**100.00**	**80.05**

Table 2. Semantic segmentation performance on Kvasir and CVC-ClinicDB datasets.

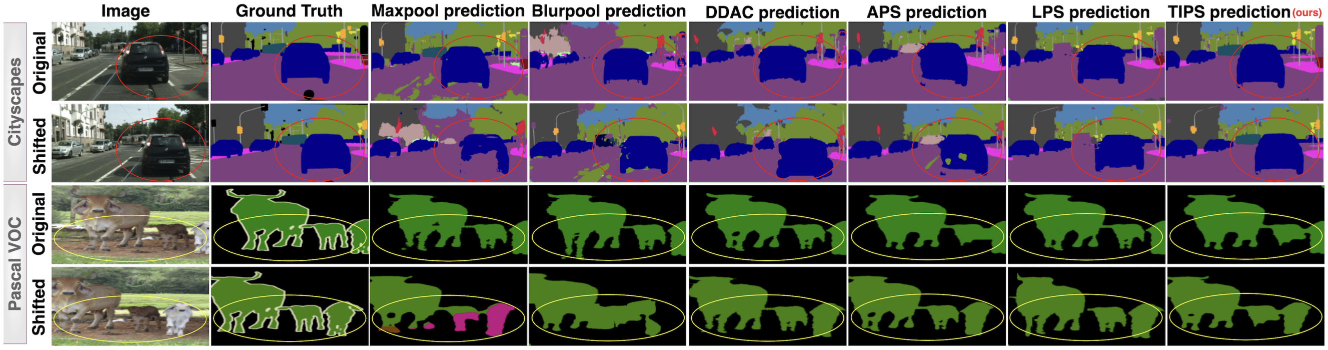

Figure 4. Qualitative comparison of segmentation masks predicted on original and shifted images. Images from Cityscapes, Pascal VOC are standard-shifted by (43,-17), (-38,0) respectively. Regions where TIPS achieve improvements (i.e. consistent segmentation quality) under linear shifts are highlighted with circles.

using a shared convolution layer and gumble softmax.

- *In TIPS*, the shared small convolution layer differs in design (Figure 2 in manuscript) from LPS. In TIPS layers, we use convolution kernels, Global Average Pooling (GAP) layer and softmax activation that learns mixing coefficients (eqn 2 in manuscript) to sample polyphase components avoiding the sensitivity to gumble softmax temperature.

1.5. More Results on Image Classification and Semantic Segmentation

Table 1 contains results from image classification experiments on Food-101 and Oxford-102 datasets. Table 2 contains quantitative results from semantic segmentation experiments on Kvasir and CVC-ClinicDB datasets while Figure 4 contains qualitative results from semantic segmentation experiments on Pascal VOC and Cityscapes datasets.

1.6. Experimental Setup for MSB - Shift Invariance Correlation Study

Table 3 shows the list of CNN architectures (including Mobile Net (2)), datasets and pooling methods that we use to obtain a total of 576 configurations for the *MSB-shift invariance* correlation study. In our study, we train each combination of architecture and dataset on 9 pooling methods: Global Average Pooling before classification with no spatial

Image Classification Experiments			**Semantic Segmentation Experiments**		
Model	**# Layers**	**Dataset**	**Model**	**# Layers**	**Dataset**
MobileNet	$\{2,3,4,5\}$	CIFAR-10	DeepLabV3+ (ResNet-18)	$\{2,3,4,5\}$	PASCAL VOC 2012
ResNet-18	$\{2,3,4,5\}$	CIFAR-100	DeepLabV3+ (ResNet-101)	$\{3,4,5.6\}$	Cityscapes
ResNet-34	$\{2,3,4,5\}$	Food-101	U-Net (ResNet-18)	$\{2,3,4,5\}$	Kvasir
ResNet-101	$\{2,3,4,5\}$	Oxford-102	U-Net (ResNet-34)	$\{2,3,4,5\}$	CVC-ClinicDB

Table 3. List of CNN architectures and datasets, tested on each pooling method for correlation analysis between MSB and Shift Invariance.

	CIFAR-10				**CIFAR-100**				**Food-101**				**Oxford-102**			
Model	$h \times w$	b	s	N	$h \times w$	b	s	N	$h \times w$	b	s	N	$h \times w$	b	s	N
MobileNet	32×32	64	60	220	32×32	64	60	220	200×200	128	60	220	200×200	128	60	220
ResNet-18	32×32	64	50	250	32×32	64	50	250	224×224	64	50	250	224×224	64	50	250
ResNet-34	32×32	64	50	250	32×32	64	50	250	224×224	64	50	250	224×224	64	50	250
ResNet-101	32×32	64	180	480	32×32	64	180	480	224×224	64	180	480	224×224	64	180	480

Table 4. Image size ($h \times w$), batch size (b), step size(s) for updating learning rate, and number of epochs (N) reported for each CNN model and image classification dataset combination for the MSB – Shift Invariance correlation analysis experiment.

downsampling of convolution features, TIPS ($\epsilon = 0.4, \alpha = 0.35$), LPS ($\tau = 0.01$), APS ($p = 2$), APS ($p \to \infty$), LPS ($\tau \to \infty$), BlurPool (LPF-5), Average Pool (2×2), and MaxPool (2×2). Furthermore, in each of the aforementioned settings, we use different number of pooling layers as shown in Table 3. While training with Global Average Pooling, we use 4 different kernel sizes ($2 \times 2, 3 \times 3, 4 \times 4, 5 \times 5$) in the first convolution layer with *same padding* to create 4 variants since varying the number of pooling layers is not possible in this setting barring that we downsample only once (downsampling the very last convolution features with Global Average Pooling before classification/segmentation layer). Furthermore, to consider a wide variety of CNN design strategies in our correlation study, we use ResNets which has architectural choices such as residual / skip connections with varying depth and MobileNet which contains group (depthwise and separable) convolutions. Additionally, to make our correlation study more robust we consider more diverse configurations such as number of pooling layers, different datasets with varying magnitude of image resolution, number of classes etc. Moreover, we train and test all of these 576 configurations which is computationally expensive while using other CNN architectures such as VGG-16 (9), ConvNext (7).

In Table 4, Table 5 we include training details such as image size, batch size, step size, number of training epochs for all model - dataset combinations used in the MSB - shift invariance correlation framework for both image classification and semantic segmentation. As discussed in Section 4 (manuscript), using Global Average Pooling with no spatial downsampling of the convolution features leads to increased computation with larger spatial features. In Table 6, we summarize a detailed analysis on how Global Average Pooling increases computational complexity in comparison to baseline MaxPool. The reported CUDA time is in *nanoseconds (ns)*, CUDA memory is in *Mega Bytes (MB)*, GFLOPs is *billions* of floating point operations per second. In (Figure 4, manuscript), we observe that Global Average Pooling improves shift invariance and reduces MSB, and Table 6 reveals that this performance gain comes at a significantly higher computational cost which is impractical. However, with TIPS we achieve comparable shift invariance and MSB by introducing marginal computational complexity in comparison to Global Average Pooling.

1.7. Experimental Setup for Image Classification, Object Detection, and Semantic Segmentation

We benchmark the performance of TIPS and prior work on five image classification datasets which are described in Table 7. We benchmark the performance of TIPS and prior work on four semantic segmentation datasets which are described in Table 8. Table 7, 8 contains further training details on all the reported datasets such as batch size, step size, number of training epochs, image/crop size, number of classes and number of images in the dataset.

The values of $\epsilon = 0.4$, $\alpha = 0.35$ were obtained using hyperparameter search on CIFAR-10. Note that we only run hyperparameter (ϵ, α) tuning on CIFAR-10 (a small dataset) and then use the same ϵ, α for other image classification, object detection, and semantic segmentation benchmarks without any hyperparameter search on other tasks or datasets. We chose $\epsilon = 0.4$ because introducing $\mathcal{L}_{undo}$ after 40% of the training duration yields the best performance (see Fig 8 in manuscript).

	Pascal VOC 2012				Cityscapes				Kvasir				CVC-ClinicDB			
Model	$h \times w$	b	s	N	$h \times w$	b	s	N	$h \times w$	b	s	N	$h \times w$	b	s	N
DeepLabV3+ (ResNet-18)	200×300	12	120	450	200×200	12	120	450	200×200	12	60	450	200×300	8	45	450
DeepLabV3+ (ResNet-101)	200×300	8	120	380	200×200	12	120	380	200×200	12	60	380	200×300	8	45	380
U-Net (ResNet-18)	200×300	12	120	180	200×200	16	120	180	200×200	16	60	180	200×300	12	45	180
U-Net (ResNet-34)	200×300	12	120	150	200×200	12	120	150	200×200	12	60	150	200×300	8	45	150

Table 5. Image size ($h \times w$), batch size (b), step size(s) for updating learning rate, and number of epochs (N) reported for each CNN model and semantic segmentation dataset combination for the MSB – Shift Invariance correlation analysis experiment.

Architecture	Pooling	CUDA Time ↓	CUDA Memory ↓	GFLOPs ↓
MobileNet	MaxPool	0.635	58.122	2.270
	TIPS	1.045	101.214	3.005
	GAP	66.609	6390.284	639.259
ResNet-18	MaxPool	1.135	21.860	4.017
	TIPS	3.525	292.844	41.937
	GAP	72.460	1957.691	1124.032
ResNet-34	MaxPool	1.954	31.904	8.128
	TIPS	5.623	334.754	71.532
	GAP	141.075	3451.912	2250.287
ResNet-101	MaxPool	4.921	131.035	31.197
	TIPS	12.204	816.791	146.596
	GAP	534.514	21144.011	8508.809

(a) Image Classification

Architecture	Pooling	CUDA Time ↓	CUDA Memory ↓	GFLOPs ↓
DeepLabV3+(ResNet-18)	MaxPool	1.33	25.216	9.570
	TIPS	3.728	380.146	91.146
	GAP	121.029	2453.834	1926.592
DeepLabV3+(ResNet-101)	MaxPool	7.52	144.737	51.447
	TIPS	18.274	911.845	246.947
	GAP	741.568	21671.086	11521.953
U-Net(ResNet-18)	MaxPool	2.754	78.574	23.567
	TIPS	8.675	1113.227	235.797
	GAP	143.402	3137.765	2700.095
U-Net(ResNet-34)	MaxPool	3.179	88.707	27.678
	TIPS	9.045	957.965	246.352
	GAP	202.312	4631.986	3826.350

(b) Semantic Segmentation

Table 6. GPU resources (CUDA time, memory, GFLOPs) allocated to convolution operations in CNNs while using different pooling operators for various CNN architectures. We observe that, performing Global Average Pooling (GAP) on the final convolution feature with no prior downsampling drastically increases GPU resources in comparison to baseline MaxPool. TIPS require additional convolution layers (Figure 2, manuscript), since it is a learnable pooling operator. Compared to MaxPool, the overhead in GPU resources with TIPS is remarkably smaller than it is for Global Average Pooling.

1.8. Computational Overhead in TIPS

Table 9 shows the percentage of additional parameters required to use TIPS on image classification and segmentation CNN models with different pooling methods and CNN architectures, for RGB images of size 224 × 224 and a batch-size of 64. TIPS introduces marginal computational overhead while still being computationally cheaper than existing pooling operators for shift invariance, i.e. DDAC. Moreover, in Table 10 we show the number of trainable parameters with different pooling operators for all the image classification, semantic segmentation CNN models. While TIPS requires higher number of trainable parameters than LPS, it is still much less than DDAC.

1.9. Effect of training on $\mathcal{L}_{FM}$

In Figure 5, we train ResNet-101 on Tiny ImageNet with TIPS and $\mathcal{L}_{FM}$ and compare it with baselines LPS, APS and MaxPool in terms of standard fidelity and MSB. To further inspect the effect of training TIPS with $\mathcal{L}_{FM}$, we train with three different setting of TIPS: (1) TIPS with $\mathcal{L}_{FM}$: to discourages both skewed and uniform τ, (2) TIPS with only the first term in $\mathcal{L}_{FM}$: to discourages skewed τ only, and (3) TIPS with only second term in $\mathcal{L}_{FM}$: to discourages uniform τ only. We observe that training TIPS with both terms from $\mathcal{L}_{FM}$ yields the maximum gain in shift fidelity and decreases MSB the most. TIPS with $\mathcal{L}_{FM}$ also outperforms other pooling methods: LPS, APS and MaxPool in terms of standard shift fidelity and MSB.

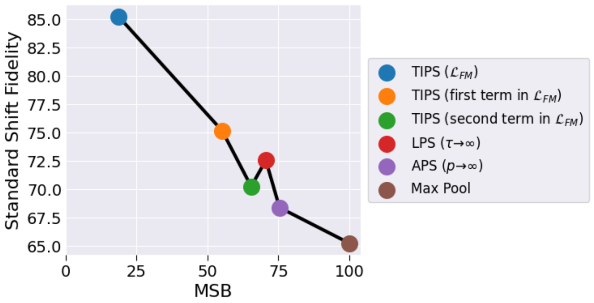

Figure 5. The effect of $\mathcal{L}_{FM}$ on TIPS is visualized by plotting standard shift fidelity versus MSB for models trained on Tiny ImageNet. Training TIPS with $\mathcal{L}_{FM}$ yields the maximum standard shift fidelity and minimum MSB.

1.10. Studying Shift Invariance on CNN architectures beyond ResNets

Tables 11, 12, contain results from DenseNet (3) and EfficientNet (11) on CIFAR-10 with different pooling methods including TIPS. We use LPF-5 for antialiasing and hyperparameters used in DenseNet and EfficientNet respec-

	Image Classification Experiments							
Dataset	**Model**	**Batch Size**	**Step Size**	**Epochs**	**Image Size**	**# Classes**	**# Training Samples**	**# Validation Samples**
CIFAR-10	ResNet-18	64	50	250	32×32	10	50,000	10,000
Food-101	ResNet-50	64	25	80	224×224	101	75,750	25,250
Oxford-102	ResNet-50	64	20	70	224×224	102	2,060	6,129
Tiny ImageNet	ResNet-101	64	180	480	64×64	200	100,000	10,000
ImageNet	ResNet-101	64	30	90	224×224	1000	1,281,167	50,000

Table 7. Training details, dataset statistics for all five datasets in our image classification experiments. Training details include batch size, step size for updating learning rate, number of training epochs, image size and dataset statistics include number of classes, training samples, validation samples.

	Semantic Segmentation Experiments							
Dataset	**Model**	**Batch Size**	**Step Size**	**Epochs**	**Image Size**	**# Classes**	**# Training Samples**	**# Validation Samples**
PASCAL VOC 2012	DeepLabV3+(ResNet-18)	12	120	450	200×300	20	1,464	1,456
Cityscapes	DeepLabV3+(ResNet-101)	12	120	380	200×200	19	2,975	500
Kvasir	UNet(ResNet-18)	12	60	180	200×200	2	850	150
CVC-ClinicDB	UNet(ResNet-34)	8	45	150	200×300	2	521	91

Table 8. Training details, dataset statistics for all four datasets in our semantic segmentation experiments. Training details include batch size, step size for updating learning rate, number of training epochs, image size and dataset statistics include number of classes, training samples, validation samples.

tively. We observe improved shift invariance with TIPS independent of CNN architecture.

1.11. Studying Effect of Normalization Layers on Shift Invariance

Using normalization layers in CNNs positively impact visual recognition performance (1; 5; 8; 12). However, the goal of this study is to carefully analyze (and isolate) the impact of pooling operators on shift invariance. While layer normalization is not the focus of this work, we have experimented on it's different alternatives and show results in Tables 13, 14. Tables 13 (batch size 32), 14 (batch size 256) contain results on CIFAR-10 with a ResNet-18 backbone with TIPS and MaxPool pooling with Batch Norm (5), Layer Norm (1), Group Norm (12), and Kernel Norm (8). We observe that usage of normalization layers leads to mixed results – this points to normalization not being a major factor for shift invariance. However, Tables 13, 14 reveal that using TIPS instead of baseline MaxPool improves shift invariance regardless of layer normalization choice.

Method	ResNet-18	ResNet-34	ResNet-50	ResNet-101
BlurPool	0.00	0.00	0.00	0.00
DDAC	7.92	10.53	9.27	4.30
APS	0.00	0.00	0.00	0.00
LPS	1.03	2.24	1.93	1.05
TIPS	5.51	4.56	2.17	3.19

(a) Image Classification

Method	DeepLabV3+(A)	DeepLabV3+(B)	UNet
BlurPool	0.00	0.00	0.00
DDAC	12.00	4.83	12.83
APS	0.00	0.00	0.00
LPS	4.40	3.25	4.79
TIPS	7.24	4.04	5.76

(b) Semantic Segmentation

Table 9. Percentage of additional parameters required by different pooling operators in comparison to MaxPool on each CNN architecture for classification and semantic segmentation. We observe that, while TIPS require more parameters than LPS, DDAC causes the maximum increase in trainable parameters *w.r.t.* baseline MaxPool.

Method	ResNet-18	ResNet-34	ResNet-50	ResNet-101
MaxPool	11.884	21.282	23.521	42.520
BlurPool	11.884	21.282	23.521	42.520
DDAC	12.825	23.524	25.701	44.349
APS	11.884	21.282	23.521	42.520
LPS	12.006	21.759	23.975	42.966
TIPS	12.539	22.253	24.031	43.876

(a) Image Classification

Method	DeepLabV3+(A)	DeepLabV3+(B)	UNet
MaxPool	20.131	58.630	7.762
BlurPool	20.131	58.630	7.762
DDAC	22.547	61.459	8.758
APS	20.131	58.630	7.762
LPS	21.017	60.536	8.134
TIPS	21.589	60.999	8.209

(b) Semantic Segmentation

Table 10. Number of trainable parameters in **Million** for various pooling methods reported for: ResNet-18, ResNet-34, ResNet-50, ResNet-101 backbones (image classification), DeepLabV3+ (A: ResNet-18, B: ResNet-101) and UNet (semantic segmentation). Number of trainable parameters are computed assuming an RGB input image of size 224 × 224.

		Standard Shift		Circular Shift	
Method	**Accuracy ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**
MaxPool	96.37	90.02	86.72	92.41	89.05
BlurPool	96.74	92.57	89.51	94.07	90.96
APS	97.27	93.31	90.82	**100.00**	97.27
LPS	97.12	94.36	91.62	**100.00**	97.12
TIPS	**97.43**	**96.71**	**94.19**	**100.00**	**97.43**

Table 11. Image classification performance on CIFAR-10 with DenseNet-BC (k=24) (3).

		Standard Shift		Circular Shift	
Method	**Accuracy ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**
MaxPool	98.90	89.14	88.13	92.19	91.22
BlurPool	98.90	91.06	90.07	92.37	91.39
APS	98.53	92.30	90.95	100.00	98.53
LPS	98.93	93.47	92.44	100.00	98.93
TIPS	**98.93**	**93.67**	**92.67**	**100.00**	**98.93**

Table 12. Image classification performance on CIFAR-10 with EfficientNet-B7 (11).

		Standard Shift		Circular Shift	
Normalization	**Accuracy ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**
Batch Norm (5)	96.02/91.43	**98.61**/87.43	**94.69**/79.94	**100.00**/90.18	96.02/82.45
Layer Norm (1)	93.43/92.25	97.34/**89.37**	90.95/**89.77**	**100.00**/90.61	93.43/83.60
Group Norm (12)	94.79/89.04	95.82/82.37	90.84/73.34	**100.00/93.59**	94.79/83.33
Kernel Norm (8)	**96.18/95.72**	98.07/86.12	94.31/82.43	**100.00**/90.81	**96.18**/86.95

Table 13. Inspecting the influence of different layer normalization strategies on ResNet-18 for CIFAR-10 with different pooling operators. All results are reported as TIPS/MaxPool with batch size of 32.

		Standard Shift		Circular Shift	
Normalization	**Accuracy ↑**	**Consistency ↑**	**Fidelity ↑**	**Consistency ↑**	**Fidelity ↑**
Batch Norm (5)	**94.71**/90.87	97.29/**88.29**	92.15/80.21	**100.00**/84.91	**94.71**/76.89
Layer Norm (1)	94.67/91.19	96.43/82.13	91.29/74.91	**100.00**/89.02	94.67/81.24
Group Norm (12)	94.14/94.02	96.80/84.82	91.14/79.38	**100.00**/92.30	94.14/86.69
Kernel Norm (8)	94.58/**94.58**	**97.44**/86.36	92.16/81.69	**100.00/93.47**	94.58/**88.43**

Table 14. Inspecting the influence of different layer normalization strategies on ResNet-18 for CIFAR-10 with different pooling operators. All results are reported as TIPS/MaxPool with batch size of 256.

References

[1] JL Ba. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 6, 7

[2] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5, 7

[4] Muhammad Hussain. Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo. *IEEE Access*, 12:42816–42833, 2024. 1

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 6, 7

[6] Jiayu Jiao, Yu-Ming Tang, Kun-Yu Lin, Yipeng Gao, Andy J Ma, Yaowei Wang, and Wei-Shi Zheng. Dilateformer: Multi-scale dilated transformer for visual recognition. *IEEE Transactions on Multimedia*, 25:8906–8919, 2023. 1

[7] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 4

[8] Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Daniel Rueckert, and Georgios Kaissis. Kernel normalized convolutional networks. *Transactions on Machine Learning Research*. 6, 7

[9] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015. 4

[10] Akito Takeki, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Parallel grid pooling for data augmentation. *arXiv preprint arXiv:1803.11370*, 2018. 1

[11] Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 5, 7

[12] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 6, 7

[13] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 1

[14] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019. 1