Contents of the Appendix

The following sections are included in the appendix:

- A review of the Π-Nets is in sec. A.
- An alternative parameterization to $\mathcal{R}$-PolyNets is introduced in sec. B.
- A number of auxiliary tables and visualizations that could not fit in the main paper are in sec. C.
- Lastly, a number of additional experiments are conducted in sec. D.

A. Background: Π-Nets

Π-Nets is a family of architectures that are high-degree polynomial expansions [5]. To reduce the parameters and enable the implementation of the polynomial expansion, coupled tensor decompositions are utilized. This results in a simple recursive formulation that enables an arbitrary degree of expansion. For instance, the N^{th} degree polynomial used for image recognition is expressed as:

$$\boldsymbol{y}_1 = \left(\boldsymbol{H}_{[1]}^T \boldsymbol{z}\right) * \left(\boldsymbol{K}_{[1]}^T \boldsymbol{k}_{[1]}\right), \tag{4}$$

$$\boldsymbol{y}_n = \left(\boldsymbol{H}_{[n]}^T \boldsymbol{z}\right) * \left(\boldsymbol{J}_{[n]}^T \boldsymbol{y}_{n-1} + \boldsymbol{K}_{[n]}^T \boldsymbol{k}_{[n]}\right) + \boldsymbol{y}_{n-1}, \tag{5}$$

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{y}_N + \boldsymbol{\theta}, \tag{6}$$

for $n = 2, \ldots, N$. The symbol $\boldsymbol{z}$ is the input vector of the polynomial, $\boldsymbol{y}$ is the output. The parameters $\boldsymbol{B}, \boldsymbol{\theta}, \{\boldsymbol{H}_{[n]}, \boldsymbol{J}_{[n]}, \boldsymbol{K}_{[n]}, \boldsymbol{k}_{[n]}\}_{n=1}^N$ are trainable. The aforementioned models can be used both in a hybrid setting (i.e., using polynomial expansion with element-wise activation) functions or as polynomial expansions. In the latter case, it was reported that despite the training accuracy reaching 100%, the testing accuracy was reduced when compared to DNNs.

B. CCP-equivalent for the regularized model

Beyond the aforementioned model of sec. 3.1, by changing the assumptions behind the tensor decomposition, one could retrieve another architecture. In this section, we demonstrate how an alternative parametrization, called CCP in [5] can be reformulated in our context. A coupled CP decomposition (CCP) can be used for tensor parameters of PNs. The recursive equation of CCP can be expressed as:

$$\boldsymbol{y}_1 = \boldsymbol{H}_{[1]}^T \boldsymbol{z},$$

$$\boldsymbol{y}_n = \left(\boldsymbol{H}_{[n]}^T \boldsymbol{z}\right) * \boldsymbol{y}_{n-1} + \boldsymbol{y}_{n-1},$$

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{y}_N + \theta,$$

for $n = 1, \ldots, N$, the parameters $\boldsymbol{B} \in \mathbb{R}^{o \times r}$, $\boldsymbol{H}_{[n]} \in \mathbb{R}^{d \times r}$ for $n = 1, \ldots, N$ are trainable.

After introducing regularization matrix, $\boldsymbol{\Phi} \in \mathbb{R}^{r \times r}$, the modified recursive relationships for $n = 2, \ldots, N$ can be expressed as follows:

$$x_n = \left(\boldsymbol{\Phi}\boldsymbol{H}_{[n]}^T \boldsymbol{z}\right) * \boldsymbol{y}_{n-1} + \boldsymbol{y}_{n-1}.$$

A schematic assuming a third order expansion ($N = 3$) is illustrated in Fig. 7.

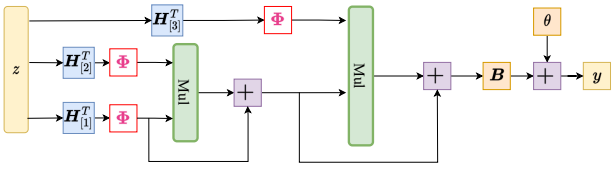


Figure 7. Schematic illustration of the regularized CCP (for third degree approximation). Symbol $*$ refers to the Hadamard product.

C. Auxiliary tables and visualizations for experiments on the main paper

Below we list the settings for experiments on the main paper in the Table 9, Table 10 and Table 11. The Table 12 ablates the accuracy for different degree polynomials on Cifar-10 and Cifar-100.

D. Additional experimental results

The following additional experimental results are added below:

1. We evaluate the classification under limited training data in sec. D.1.
2. We conduct an error analysis for best and worst performing classes in sec. D.2.
3. In sec. D.3, an comparison with convolutional kernel networks is conducted.

Below, we also add details on the datasets used in this paper:

Datasets: The following datasets are used in our evaluation:

1. *Cifar-10* [22] is a popular image recognition dataset consisting of $50,000$ training and $10,000$ testing images evenly distributed across 10 classes. Each image is of resolution 32×32.
2. *Cifar-100* [23] includes images similar to Cifar-10. Cifar-100 contains 100 object classes with 600 (500 for training, 100 for testing) images annotated per class.

Table 9. Experimental settings in sec 5.2 and sec 5.3. Note the hyper-parameters of label smoothing are selected on the validation sets of Cifar-10 and Cifar-100.

	Cifar-10/Cifar-100/STL-10/Tiny ImageNet	ImageNet
optimizer	SGD	SGD
base learning rate	$1e-1$	$1e-1$
weight decay	$5e-4$	$1e-4$
optimizer momentum	0.9	0.9
batch size	128 (64: Tiny ImageNet)	256
training epochs	120	100
learning rate schedule	multi-step decay exponential decay: $\mathcal{R}$-PolyNets/ $\mathcal{D}$-PolyNets (Tiny ImageNet)	multi-step decay
label smoothing: $\mathcal{R}$-PolyNets/$\mathcal{D}$-PolyNets	0.1: (Cifar-10 and STL-10) 0.4: (Cifar-100) 0.6: (Tiny ImageNet)	0.1

Table 10. Experimental setting in sec 5.4.

	Speech Command
optimizer	SGD
base learning rate	$1e-1$
weight decay	$5e-4$
optimizer momentum	0.9
batch size	128
training epochs	120
learning rate schedule	multi-step decay
label smoothing: $\mathcal{R}$-PolyNets/$\mathcal{D}$-PolyNets	0.1

Table 11. Experimental setting in sec 5.5.

	Oxford 102 Flowers
optimizer	SGD
base learning rate	$1e-1$
weight decay	$5e-4$
optimizer momentum	0.9
batch size	64
training epochs	120
learning rate schedule	multi-step decay
label smoothing: $\mathcal{R}$-PolyNets/$\mathcal{D}$-PolyNets	0.4

3. *STL-10* [7] contains 10 object classes that are similar to Cifar-10. Each image is of resolution 96×96, while the dataset contains $5,000$ images. This dataset is used to evaluate the performance on images of higher resolution, while using limited data.

4. *Tiny ImageNet* [25] contains 200 object classes, where each image is of resolution 64×64. There are 500 images annotated per class, while the object classes demonstrate a larger variance than the aforementioned datasets.

5. *Speech Commands dataset* [46] includes $60,000$ audio files; each audio contains a single word of a duration of one second. There are 35 different words (classes) with each word having $1,500 - 4,100$ recordings. Every audio file is converted into a mel-spectrogram of resolution 32×32.

6. *ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)* [10] contains over one million training images and $50,000$ validation images from $1,000$ object classes. Each image depicts natural scenes and is annotated with a single object per image.

Table 12. Accuracy of $\mathcal{R}$-PolyNets with varying degree polynomials on Cifar-10 and Cifar-100. Each block is a degree 2 polynomial expansion, which results in the 2^6 expansion if we add 6 such blocks. Blocks with higher-degree can also be used, however we note that training those has not been as stable in our experience.

Dataset	**Degree polynomials**	**Accuracy**
Cifar-10	2^2 degree expansion	0.880 ± 0.003
	2^4 degree expansion	0.924 ± 0.003
	2^6 degree expansion	0.931 ± 0.001
	2^8 degree expansion	0.945 ± 0.000
	2^{10} degree expansion	$\mathbf{0.950 \pm 0.002}$
Cifar-100	2^2 degree expansion	0.671 ± 0.003
	2^4 degree expansion	0.732 ± 0.002
	2^6 degree expansion	0.738 ± 0.002
	2^8 degree expansion	0.769 ± 0.002
	2^{10} degree expansion	$\mathbf{0.775 \pm 0.002}$

D.1. Image classification with limited data

We conduct an experiment on Cifar-10 in the presence of limited data. The hyper-parameters in sec. 5.2 are used unchanged, while only the number of training samples of each class is reduced. The results in Table 13 exhibit that $\mathcal{R}$-PolyNets outperform Π-Nets in the presence of limited training data. Notice that in the extreme case of only 50 samples per class, there is a relative increase of 50% from the accuracy of Π-Nets. The goal of this experiment is to explore how Π-Nets and $\mathcal{D}$-PolyNets perform in the presence

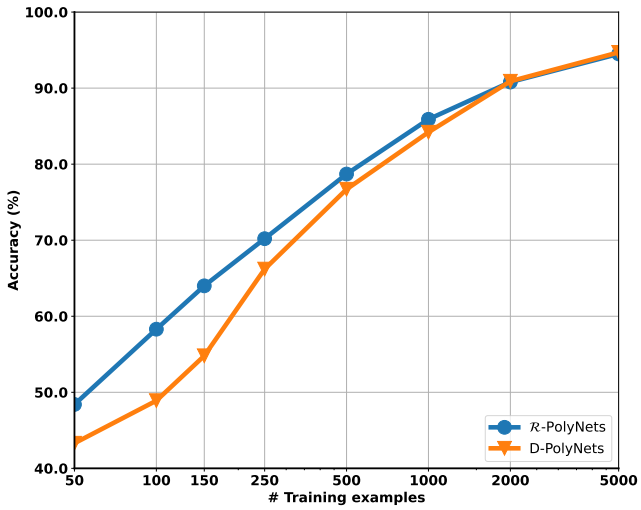


Figure 8. Image classification with limited data on Cifar-10. The x-axis declares the number of training samples per class (log-axis).

of limited data. Indeed, Fig. 8 confirms that both networks perform reasonably in the case of limited data.

Table 13. Accuracy of image classification with limited data on Cifar-10. Note that $\mathcal{R}$-PolyNets without activation functions can outperform Π-Nets without activation functions significantly on limited data of Cifar-10.

Training samples per class	Π-Nets	$\mathcal{R}$-PolyNets
50	0.314 ± 0.005	$\mathbf{0.484 \pm 0.004}$
100	0.355 ± 0.010	$\mathbf{0.583 \pm 0.003}$
150	0.396 ± 0.010	$\mathbf{0.640 \pm 0.006}$

D.2. Error Analysis

We use our best-performing $\mathcal{R}$-PolyNets to calculate per-class error rates for all 200 classes on the validation dataset of a large-scale classification dataset, Tiny ImageNet [25]. We report the top-5 accurate and misclassified classes in Table 14. Also, we present the images of the most accurate class (king penguin) and the most misclassified class (umbrella) in Fig. 9.

Remarkably, $\mathcal{R}$-PolyNets achieve above 85% validation accuracy for the top-5 accurate classes. We analyse the images of the most accurate class and misclassified class. As shown in Fig. 9, the king penguins occupy most regions in the images. Also, they have similar shape, color and texture. On the other side, the umbrellas in Fig. 9 have different colors and shapes. Furthermore, the images are dominated by the other objects such as human beings and landscape. The saliency maps in Fig. 10 computed by GradCAM [38] indicate $\mathcal{R}$-PolyNets can concentrate on the main object in an image. By comparison, Π-Nets recognize the lesser panda at 92.7% accuracy, but the saliency maps in Fig. 10 show Π-Nets do not concentrate on the main object in an image.

Table 14. Top-5 Accurate/Misclassified Classes on Tiny ImageNet. Note that $\mathcal{R}$-PolyNets can achieve above 85% validation accuracy for the top-5 accurate classes.

Class Name	Accuracy	Class Name	Accuracy
king penguin	0.902	backpack	0.358
lesser panda	0.900	bucket	0.327
sea slug	0.895	plunger	0.296
bullet train	0.882	wooden spoon	0.278
Persian cat	0.879	umbrella	0.243

(a) Most accurate class (90.2% accuracy) (b) Most misclassified class (24.3% accuracy)

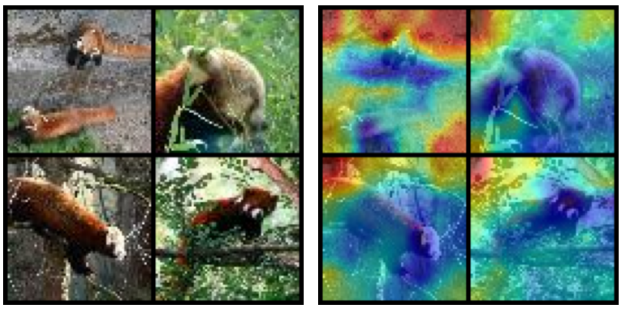
Figure 9. Images of the Most Accurate/Misclassified Class recognized by $\mathcal{R}$-PolyNets. As in Fig 9, the king penguins in (a) have similar characteristics, and occupy most regions in the images. On the other side, the images in (b) are dominated by other objects such as persons and landscape.

D.3. Comparison with convolutional kernel networks

We conduct experiments to compare $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets with supervised convolutional kernel networks (SCKNs) [29], which is among the principled design choices. The results of $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets are the same as those in the main paper, i.e., in Table 3. The accuracy for each model is reported in Table 15. Notice that the proposed $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets outpuperform the newly added baseline.

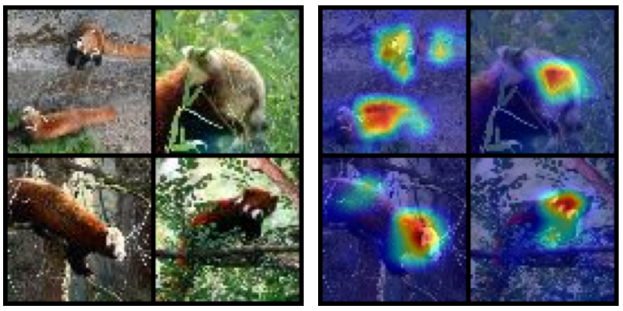
D.4. Regularized PDC, ResNext and dense connections for PDC

To showcase the representative power of the regularized polynomial expansion and dense connections across different polynomial nets, we firstly apply the proposed regularization schemes (IBN + max pooling + Dropblock + Label smoothing) in the influential ResNext [48] and the recent PDC [4]. The regularized ResNext is called $\mathcal{R}$-PolyNeXt,



(a) Lesser panda (Π-Nets: 92.7% accuracy)

(b) saliency maps of Π-Nets



(c) Lesser panda ($\mathcal{R}$-PolyNets: 90.0% accuracy)

(d) saliency maps of $\mathcal{R}$-PolyNets

Figure 10. Saliency maps of Π-Nets and $\mathcal{R}$-PolyNets. As in Fig 10, Π-Nets can recognize the lesser panda in (a) at 93% accuracy, but the saliency maps in (b) indicate Π-Nets can not concentrate on the main object in an image. The saliency maps in (d) indicate $\mathcal{R}$-PolyNets can concentrate on the main object in an image.

Table 15. Accuracy on Cifar-10, Cifar-100, STL-10 and Tiny ImageNet. The symbol '# par' abbreviates the number of parameters. $\mathcal{D}$-PolyNets containing 7M parameters. Note that $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets without activation functions can outperform SCKNs on Cifar-10, Cifar-100, STL-10 and Tiny ImageNet by a large margin.

Dataset	Model	# par	Accuracy
Cifar-10	SCKNs	3.4M	0.895 ± 0.002
	$\mathcal{R}$-PolyNets	11.9M	0.945 ± 0.000
	$\mathcal{D}$-PolyNets	7.1M	$\mathbf{0.947 \pm 0.002}$
Cifar-100	SCKNs	3.5M	0.610 ± 0.003
	$\mathcal{R}$-PolyNets	11.9M	$\mathbf{0.769 \pm 0.002}$
	$\mathcal{D}$-PolyNets	7.2M	0.767 ± 0.003
STL-10	SCKNs	3.4M	0.527 ± 0.012
	$\mathcal{R}$-PolyNets	11.9M	0.828 ± 0.003
	$\mathcal{D}$-PolyNets	7.1M	$\mathbf{0.834 \pm 0.006}$
Tiny ImageNet	SCKNs	4.2M	0.409 ± 0.001
	$\mathcal{R}$-PolyNets	12.0M	0.615 ± 0.004
	$\mathcal{D}$-PolyNets	7.2M	$\mathbf{0.618 \pm 0.001}$

while the regularized PDC is called $\mathcal{R}$-PDC. The results

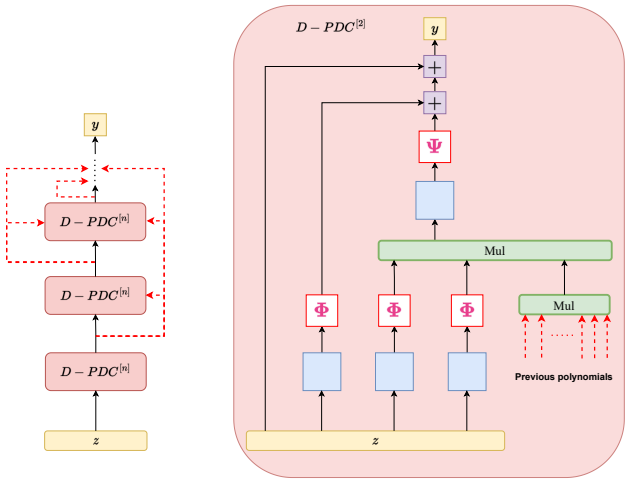


Figure 11. Schematic illustration of $\mathcal{D}$-PDC. On the left the overall structure is presented, while on the right a single second-degree polynomial using the structure of $\mathcal{D}$-PDC is visualized. The red arrows depict the newly added connections with respect to previous polynomial expansions.

for ResNext are reported in Table 18. Even though $\mathcal{R}$-PolyNeXt performs on par with ResNext, we notice that there is some training instability that did not emerge in regularizing PDC or Π-Nets. It is possible that further tuning is required for converting more complex models, such as ResNext, into polynomial expansions.

Furthermore, we also enable additional skip connections across polynomials for PDC. The new type of PDC is called $\mathcal{D}$-PDC. The schematic in Fig. 11 depicts $\mathcal{D}$-PDC assuming each polynomial includes a single recursive step. This can be trivially extended to any number of recursive steps, while each polynomial can also rely on a different tensor decomposition. The same regularization scheme (IBN + max pooling + Dropblock + Label smoothing) in $\mathcal{D}$-PolyNets is used in $\mathcal{D}$-PDC. The accuracy for each model is reported in Table 16. Notice that the $\mathcal{R}$-PDC and $\mathcal{D}$-PDC both outperform the PDC. The rest of the patterns, e.g., $\mathcal{D}$-PDC versus $\mathcal{R}$-PDC, are similar to the experiments in the main paper.

D.5. Comparison with deeper ResNets

We conduct experiments to compare deeper $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets with deeper ResNets. The experimental settings described in sec. 5.2 remain unchanged for these comparisons. The accuracy for each model is reported in Table 19. It is noteworthy that the proposed $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets outperform ResNets when their architectures are deeper.

D.6. FLOPs

We compute the floating-point operations per second (FLOPs) for $\mathcal{R}$-PolyNets, $\mathcal{D}$-PolyNets, Π-Nets, and ResNet18 on both small datasets and ImageNet. The re-

Table 16. Accuracy on Cifar-10, Cifar-100, STL-10 and Tiny ImageNet. The symbol '# par' abbreviates the number of parameters. Note that $\mathcal{R}$-PDC and $\mathcal{D}$-PDC without activation functions can outperform PDC without activation functions significantly on Cifar-10, Cifar-100, STL-10 and Tiny ImageNet.

Dataset	Model	# par	Accuracy
Cifar-10	PDC	5.4M	0.909 ± 0.002
	$\mathcal{R}$-PDC	7.3M	0.947 ± 0.001
	$\mathcal{D}$-PDC	6.0M	$\mathbf{0.949 \pm 0.002}$
Cifar-100	PDC	5.5M	0.689 ± 0.002
	$\mathcal{R}$-PDC	7.4M	0.757 ± 0.003
	$\mathcal{D}$-PDC	6.0M	$\mathbf{0.762 \pm 0.001}$
STL-10	PDC	5.4M	0.681 ± 0.006
	$\mathcal{R}$-PDC	7.3M	0.833 ± 0.007
	$\mathcal{D}$-PDC	6.0M	$\mathbf{0.855 \pm 0.003}$
Tiny ImageNet	PDC	5.5M	0.452 ± 0.002
	$\mathcal{R}$-PDC	7.4M	0.560 ± 0.005
	$\mathcal{D}$-PDC	6.0M	$\mathbf{0.569 \pm 0.002}$

Table 17. Accuracy of $\mathcal{D}$-PolyNets without IBN and without label smoothing (mentioned as '$\mathcal{D}$-PolyNets without reg' below) on Cifar-10 and Cifar-100. The symbol '# par' abbreviates the number of parameters.

Dataset	Model	# par	Accuracy
Cifar-10	Π-Nets	11.9M	0.907 ± 0.003
	$\mathcal{D}$-PolyNets without reg	7.1M	$\mathbf{0.934 \pm 0.002}$
Cifar-100	Π-Nets	11.9M	0.677 ± 0.006
	$\mathcal{D}$-PolyNets without reg	7.2M	$\mathbf{0.726 \pm 0.006}$

Table 18. Accuracy of ResNext [48] and the corresponding $\mathcal{R}$-PolyNeXt on Cifar-10 and Cifar-100.

Dataset	Model	# par	Accuracy
Cifar-10	ResNeXt-29, 8×64d	34.4M	0.964
	$\mathcal{R}$-PolyNeXt-29, 8×64d	38.6M	**0.965**
Cifar-100	ResNeXt-29, 8×64d	34.4M	0.822
	$\mathcal{R}$-PolyNeXt-29, 8×64d	38.7M	**0.824**

sults of these computations are presented in Table 20 and Table 21. Notice that the proposed $\mathcal{R}$-PolyNets has a similar FLOP as the previously proposed Π-Nets, while $\mathcal{D}$-PolyNets has only a marginal increase in the FLOPs.

Table 19. Accuracy on Cifar-10 and Cifar-100. The symbol '# par' abbreviates the number of parameters. Note that deeper $\mathcal{R}$-PolyNets and $\mathcal{D}$-PolyNets without activation functions can outperform deeper ResNets on Cifar-10, and Cifar-100.

Dataset	Model	# par	Accuracy
Cifar-10	ResNet34	21.3M	0.947 ± 0.002
	$\mathcal{R}$-PolyNets34	22.5M	0.950 ± 0.001
	$\mathcal{D}$-PolyNets34	13.5M	$\mathbf{0.951 \pm 0.002}$
	ResNet152	58.2M	0.943 ± 0.003
	$\mathcal{R}$-PolyNets152	58.5M	0.952 ± 0.001
	$\mathcal{D}$-PolyNets152	54.2M	$\mathbf{0.953 \pm 0.002}$
Cifar-100	ResNet34	21.3M	0.762 ± 0.004
	$\mathcal{R}$-PolyNets34	22.6M	$\mathbf{0.788 \pm 0.002}$
	$\mathcal{D}$-PolyNets34	13.5M	0.787 ± 0.001
	ResNet152	58.3M	0.768 ± 0.005
	$\mathcal{R}$-PolyNets152	58.5M	$\mathbf{0.793 \pm 0.004}$
	$\mathcal{D}$-PolyNets152	54.2M	0.791 ± 0.001

Table 20. FLOPs on Cifar-10, Cifar-100, STL-10 and Tiny ImageNet.

Dataset	Model	GFLOPs
Cifar-10	ResNet18	0.56
	Hybrid Π-Nets	0.46
	Π-Nets	0.59
	$\mathcal{R}$-PolyNets	0.59
	$\mathcal{D}$-PolyNets	0.55
Cifar-100	ResNet18	0.56
	Hybrid Π-Nets	0.46
	Π-Nets	0.59
	$\mathcal{R}$-PolyNets	0.59
	$\mathcal{D}$-PolyNets	0.55
STL-10	ResNet18	5.01
	Hybrid Π-Nets	4.11
	Π-Nets	5.31
	$\mathcal{R}$-PolyNets	5.31
	$\mathcal{D}$-PolyNets	4.94
Tiny ImageNet	ResNet18	2.23
	Hybrid Π-Nets	1.83
	Π-Nets	2.36
	$\mathcal{R}$-PolyNets	2.36
	$\mathcal{D}$-PolyNets	2.19

Table 21. FLOPs on ImageNet. Notice that the proposed $\mathcal{R}$-PolyNets has a similar FLOP as the previously proposed Π-Nets.

Model	GFLOPs
ImageNet-1K trained models	
ResNet18	1.82
ResNet18 without activations	1.82
Hybrid Π-Nets	1.92
Π-Nets	1.92
$\mathcal{R}$-PolyNets	1.92
$\mathcal{D}$-PolyNets	1.98