A Detailed notation

Products: The *Hadamard* product of $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{I \times N}$ is defined as $\boldsymbol{A} * \boldsymbol{B}$ and is equal to $a_{(i,j)}b_{(i,j)}$ for the (i,j) element. The *Khatri-Rao* product of matrices $\boldsymbol{A} \in \mathbb{R}^{I \times N}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times N}$ is denoted by $\boldsymbol{A} \odot \boldsymbol{B}$ and yields a matrix of dimensions $(IJ) \times N$. The Khatri-Rao product for a set of matrices $\{\boldsymbol{A}_{[m]} \in \mathbb{R}^{I_m \times N}\}_{m=1}^{M}$ is abbreviated by $\boldsymbol{A}_{[1]} \odot \boldsymbol{A}_{[2]} \odot \cdots \odot \boldsymbol{A}_{[M]} \doteq \bigodot_{m=1}^{M} \boldsymbol{A}_{[m]}$.

Tensors: Each element of an M^{th} order tensor $\boldsymbol{\mathcal{X}}$ is addressed by M indices, i.e., $(\boldsymbol{\mathcal{X}})_{i_1,i_2,\ldots,i_M} \doteq x_{i_1,i_2,\ldots,i_M}$. An M^{th}-order tensor $\boldsymbol{\mathcal{X}}$ is defined over the tensor space $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$, where $I_m \in \mathbb{Z}$ for $m = 1, 2, \ldots, M$. The *mode-m unfolding* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ maps $\boldsymbol{\mathcal{X}}$ to a matrix $\boldsymbol{X}_{(m)} \in \mathbb{R}^{I_m \times \bar{I}_m}$ with $\bar{I}_m = \prod_{\substack{k=1 \\ k \neq m}}^{M} I_k$ such that the tensor element $x_{i_1,i_2,\ldots,i_M}$ is mapped to the matrix element $x_{i_m,j}$ where $j = 1 + \sum_{\substack{k=1 \\ k \neq m}}^{M} (i_k - 1)J_k$ with $J_k = \prod_{\substack{n=1 \\ n \neq m}}^{k-1} I_n$. The *mode-m vector product* of $\boldsymbol{\mathcal{X}}$ with a vector $\boldsymbol{c} \in \mathbb{R}^{I_m}$, denoted by $\boldsymbol{\mathcal{X}} \times_m \boldsymbol{c} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{m-1} \times I_{m+1} \times \cdots \times I_M}$, results in a tensor of order $M - 1$:

$$(\boldsymbol{\mathcal{X}} \times_m \boldsymbol{c})_{i_1,\ldots,i_{m-1},i_{m+1},\ldots,i_M} = \sum_{i_m=1}^{I_m} x_{i_1,i_2,\ldots,i_M} u_{i_m}. \tag{13}$$

The *CP decomposition* [32] factorizes a tensor into a sum of component rank-one tensors. The rank-R CP decomposition of an M^{th}-order tensor $\boldsymbol{\mathcal{X}}$ is written as:

$$\boldsymbol{\mathcal{X}} \doteq [\![\boldsymbol{C}_{[1]}, \boldsymbol{C}_{[2]}, \ldots, \boldsymbol{C}_{[M]}]\!] = \sum_{r=1}^{R} \boldsymbol{c}_r^{(1)} \circ \boldsymbol{c}_r^{(2)} \circ \cdots \circ \boldsymbol{c}_r^{(M)}, \tag{14}$$

where $\circ$ is the vector outer product. The factor matrices $\big\{\boldsymbol{C}_{[m]} = [\boldsymbol{c}_1^{(m)}, \boldsymbol{c}_2^{(m)}, \cdots, \boldsymbol{c}_R^{(m)}] \in \mathbb{R}^{I_m \times R}\big\}_{m=1}^{M}$ collect the vectors from the rank-one components. By considering the mode-1 unfolding of $\boldsymbol{\mathcal{X}}$, the CP decomposition can be written in matrix form as:

$$\boldsymbol{X}_{(1)} \doteq \boldsymbol{C}_{[1]} \Big(\bigodot_{m=M}^{2} \boldsymbol{C}_{[m]} \Big)^T \tag{15}$$

The following lemma is useful in our method:

Lemma 1 ([9]). *For a set of N matrices $\{\boldsymbol{A}_{[\nu]} \in \mathbb{R}^{I_\nu \times K}\}_{\nu=1}^{N}$ and $\{\boldsymbol{B}_{[\nu]} \in \mathbb{R}^{I_\nu \times L}\}_{\nu=1}^{N}$, the following equality holds:*

$$(\bigodot_{\nu=1}^{N} \boldsymbol{A}_{[\nu]})^T \cdot (\bigodot_{\nu=1}^{N} \boldsymbol{B}_{[\nu]}) = (\boldsymbol{A}_{[1]}^T \cdot \boldsymbol{B}_{[1]}) * \ldots * (\boldsymbol{A}_{[N]}^T \cdot \boldsymbol{B}_{[N]}) \tag{16}$$

B Polynomials as a single tensor product

As mentioned in the main paper polynomials and tensors are closely related. To illustrate the differences between the proposed variant of sec. 4.1 and the proposed taxonomy we can formulate them as a single tensor product. We assume a second-degree polynomial expansion of (1). The tensors are then up to third-order, which enables a visualization (as in the Fig.1). The initial equation is:

$$\boldsymbol{y} = \boldsymbol{\beta} + \left(\boldsymbol{W}^{[1]}\right)^T \boldsymbol{z} + \left(\boldsymbol{\mathcal{W}}^{[2]} \times_2 \boldsymbol{z} \times_3 \boldsymbol{z}\right) \tag{17}$$

The τ^{th} output of (17) can be written in element-wise form as:

$$y_\tau = \beta_\tau + \sum_{k=1}^{\delta} w^{[1]}_{\tau,k} z_k + \sum_{k,m=1}^{\delta} w^{[2]}_{\tau,k,m} z_k z_m \tag{18}$$

We can collect all the parameters of (17) under a single tensor by padding the input $\boldsymbol{z} \in \mathbb{R}^\delta$. Specifically, if we consider the padded version $\tilde{\boldsymbol{z}} = [z_1, \dots, z_\delta, 1]^T$, then (17) can be written in the format $\boldsymbol{y} = \tilde{\boldsymbol{\mathcal{W}}} \times_2 \tilde{\boldsymbol{z}} \times_3 \tilde{\boldsymbol{z}}$ as we demonstrate below.

The τ^{th} output of $\tilde{\boldsymbol{\mathcal{W}}} \times_2 \tilde{\boldsymbol{z}} \times_3 \tilde{\boldsymbol{z}}$ is:

$$\begin{aligned} y_\tau = \sum_{k,m=1}^{\delta+1} \tilde{w}_{\tau,k,m} \tilde{z}_k \tilde{z}_m = \underbrace{\tilde{w}_{\tau,\delta+1,\delta+1}}_{\text{constant term}} + \\ \underbrace{\sum_{m=1}^{\delta} \tilde{w}_{\tau,\delta+1,m} z_m + \sum_{k=1}^{\delta} \tilde{w}_{\tau,k,\delta+1} z_k}_{\text{first-degree term}} + \underbrace{\sum_{k,m=1}^{\delta} \tilde{w}_{\tau,k,m} z_k z_m}_{\text{second-degree term}} \end{aligned} \tag{19}$$

If we set:

$$\begin{cases} \beta_\tau = \tilde{w}_{\tau,\delta+1,\delta+1} & \\ w^{[1]}_{\tau,k} = \tilde{w}_{\tau,\delta+1,k} + \tilde{w}_{\tau,k,\delta+1} & \text{for } k = 1, \dots, \delta \\ w^{[2]}_{\tau,k,m} = \tilde{w}_{\tau,k,m} & \text{for } k, m = 1, \dots, \delta \end{cases} \tag{20}$$

then (19) becomes the polynomial expansion of (18).

This enables us to express different degree polynomial expansions with a third-order tensor. The first-degree methods, e.g., ResNet [22], have $w^{[2]}_{\tau,k,m} = 0$, while SENet [25] assumes $w^{[1]}_{\tau,k} = 0$. The Π-net family assumes low-rank decomposition with shared factors, i.e., the low-rank decompositions of $\boldsymbol{\mathcal{W}}^{[n]}{}_{n=1}^{N}$ share factor matrices. On the contrary, our proposed PDC does not assume a sharing pattern, thus it can express independently the terms $\boldsymbol{W}^{[1]}, \boldsymbol{\mathcal{W}}^{[2]}$.

C Proofs

Claim. The Squeeze-and-excitation block of (3) is a special form of second-degree polynomial term.

Proof. The global pooling function on a matrix $\boldsymbol{C}$ can be expressed as $\frac{1}{hw}\overrightarrow{\mathbf{1}}^T\boldsymbol{C}$. The r function that replicates the channels acts on a vector $\boldsymbol{c}$ and results in the expression $\overrightarrow{\mathbf{1}}\boldsymbol{c}^T$.

The identity $\boldsymbol{X}*\boldsymbol{a}\boldsymbol{b}^T = diag(\boldsymbol{a})\boldsymbol{X}diag(\boldsymbol{b})$ can be used to convert the Hadamard product of (3) into a matrix multiplication [55]. Then, (3) becomes:

$$\boldsymbol{Y}_s = (\boldsymbol{Z}\boldsymbol{C}_1) * \overrightarrow{\mathbf{1}}\left(\left(\frac{1}{hw}\overrightarrow{\mathbf{1}}^T\boldsymbol{Z}\boldsymbol{C}_1\right)\boldsymbol{C}_2\right)^T = (\boldsymbol{Z}\boldsymbol{C}_1)\frac{1}{hw}diag(\boldsymbol{C}_2^T\boldsymbol{C}_1^T\boldsymbol{Z}^T\overrightarrow{\mathbf{1}}) = (\boldsymbol{Z}\boldsymbol{C}_1)\frac{1}{hw}\boldsymbol{\mathcal{I}} \times_3 (\boldsymbol{C}_2^T\boldsymbol{C}_1^T\boldsymbol{Z}^T\overrightarrow{\mathbf{1}}) \tag{21}$$

where as a reminder $\boldsymbol{\mathcal{I}}$ is a third-order super-diagonal unit tensor. The last equation is a second-degree term with $\boldsymbol{\Phi}_1^{[2]}(\boldsymbol{Z}) = \boldsymbol{Z}\boldsymbol{C}_1$ and $\boldsymbol{\Phi}_2^{[2]}(\boldsymbol{Z}) = \frac{1}{hw}\boldsymbol{\mathcal{I}} \times_3 (\boldsymbol{C}_2^T\boldsymbol{C}_1^T\boldsymbol{Z}^T\overrightarrow{\mathbf{1}})$.

D Auxiliary experiments

Table 9: Image classification on CIFAR100 with variants of ResNet34.

Model	**# param** ($\times 10^6$)	**Accuracy**
ResNet34	21.3	0.769
Π-net-ResNet	14.7	0.769
PDC-channels	36.3	**0.774**
PDC	**10.5**	0.770

D.1 Image classification with limited data

A number of experiments is performed by progressively reducing the number of training samples per class. The number of samples is reduced uniformly from the original $5,000$ down to 50 per class, i.e., a $100\times$ reduction, in CIFAR10. The architectures of Table 3 (similar to ResNet18) are used unchanged; only the number of training samples is progressively reduced. The resulting Fig. 4 visualizes the performance as we decrease the training samples. The accuracy of ResNet18 decreases fast for limited training samples. SENet deteriorates at a slower pace,

steadily increasing the difference from ResNet18 (note that both share similar number of parameters). Π-net-ResNet improves upon SENet and performs better even under limited data. However, the proposed PDC-comp outperforms all the compared methods for 50 training samples per class. The difference in the accuracy between PDC and Π-net-ResNet increases as we reduce the number of training samples. Indicatively, with 50 samples per class, ResNet18 attains accuracy of 0.347, SENet scores 0.355, Π-net-ResNet scores 0.397 and PDC-comp scores 0.426, which is a 22% increase over the ResNet18 baseline.

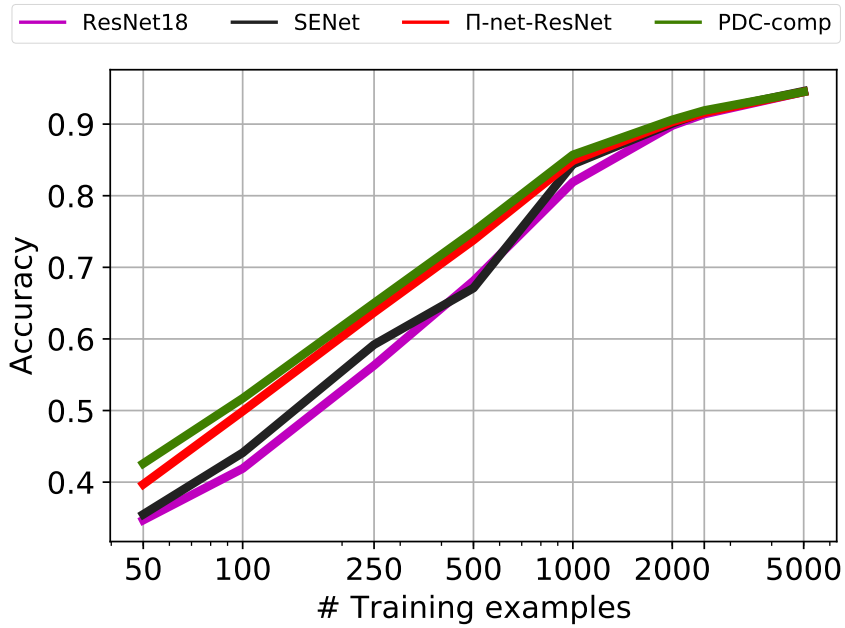

Fig. 4: Image classification with limited data. The x-axis declares the number of training samples per class (log-axis). As the number of samples is reduced (i.e., moving from right to the left), the performance gap between Π-net-ResNet and ResNet18 increases. Similarly, PDC-comp performs better than Π-net-ResNet, especially in the limited data regimes on the left.

D.2 Classification without activation functions

Typical feed-forward neural networks, such as CNNs, require activation functions to learn complex functions [23]. However, the proposed view of polynomial expansion enables capturing higher-order correlations even in the absence of activation functions. That is, the expressivity of higher-degree polynomials can be assessed without activation functions. We conduct a series of experiments on all three datasets with higher-degree polynomials. Our core experiments study

the higher-degree polynomials of Π-nets [8], versus the proposed model of (10). We also implement the ResNet without activation functions to assess how first-degree polynomials perform.

For the first experiment, we utilize ResNet18 as the backbone and test the baselines on CIFAR100. Three variations of $Pi-$net are considered as the compared methods: one with second-degree, one with third-degree and one with fourth-degree residual blocks. The same polynomial expansions are used for the proposed PDC. The accuracy of each method is reported in Table 10. All the variants of Π-net-ResNet and PDC exhibit a high accuracy based solely on the high-degree polynomial expansion. However, Π-net-ResNet saturates when the residual block is a third or fourth degree polynomial, while the PDC does not suffer from the same issue. On the contrary, the performance of the PDC variant with third and forth degree residual block outperforms the second-degree residual block.

Table 10: Image classification on CIFAR100 without activation functions. Both Π-net-ResNet and PDC use high-degree polynomial expansion to achieve high accuracy even in the absence of activation functions. The proposed PDC achieves both increased performance and improves its performance when each residual block has third or fourth degree polynomial instead of second.

Model	**# param** ($\times 10^6$)	**Accuracy**
ResNet18	11.2	0.168
Π-net-ResNet	11.9	0.667
Π-net-ResNet$^{[3]}$	11.2	0.648
Π-net-ResNet$^{[4]}$	11.2	0.626
PDC	**5.46**	0.689
PDC$^{[3]}$	11.2	**0.703**
PDC$^{[4]}$	18.8	0.699

The models are also evaluated on CIFAR10 with ResNet18 and three variants of $\Pi-$nets as the backbone. Three variants of PDC with different expansion degrees are designed. The results are tabulated on Table 11. Each variant of Π-net-ResNet and PDC surpasses the 0.87 accuracy and outperform the ResNet18 by a wide margin. In contrast to Π-net-ResNet, the performance of PDC does not decrease when the degree of the residual block increases, i.e., from second to fourth-degree. Overall, PDC outperforms $\Pi-$net.

The last experiment is conducted on the Speech Commands dataset. The baseline of ResNet18 is selected, while the Π-net-ResNet is the compared method. The results in Table 12 depict the same motif: the two polynomial expansions are very expressive. Impressively, in this dataset the result without activation functions is only 0.007 decreased when compared to the respective results with activation functions. This highlights that simple datasets might not always demand activation functions to achieve high-accuracy.

Table 11: Image classification on CIFAR10 without activation functions. The results illustrate the expressiveness of the proposed model even in the absence of activation functions. Notice that PDC$^{[3]}$ improves upon PDC with second-degree blocks. On the contrary, this does not happen to the compared Π-net-ResNet.

Model	**# param** ($\times 10^6$)	**Accuracy**
ResNet18	11.2	0.391
Π-net-ResNet	11.9	0.907
Π-net-ResNet$^{[3]}$	11.2	0.891
Π-net-ResNet$^{[4]}$	11.2	0.877
PDC	**5.4**	0.909
PDC$^{[3]}$	11.2	**0.918**
PDC$^{[4]}$	18.8	**0.918**

Table 12: Audio classification without activation functions.

Model	**# param** ($\times 10^6$)	**Accuracy**
ResNet18	11.2	0.464
Π-net-ResNet	11.9	0.971
PDC	**5.4**	**0.972**

Table 13: **COCO object detection and segmentation results** using Mask-RCNN and Cascade Mask-RCNN. The backbone models are pre-trained ResNet18 and PDC-ResNet18 models on ImageNet-1K. We employ MMDetection with 1× schedule.

backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Mask-RCNN 1× schedule						
ResNet18	33.9	53.9	36.2	31.0	50.9	33.0
PDC-ResNet18	34.8	55.2	37.4	31.8	52.2	34.1
Cascade Mask-RCNN 1× schedule						
ResNet18	37.3	54.8	40.4	32.6	52.2	34.9
PDC-ResNet18	38.1	55.9	41.7	33.2	53.3	35.7

D.3 Object detection and segmentation

We adopt MS COCO 2017 [39] as the primary benchmark for the experiments of object detection and segmentation. We use the train split (118k images) for training and report the performance on the val split (5k images). We employ standard evaluation metrics for COCO dataset, where multiple IoU thresholds from 0.5 to 0.95 are applied. The detection results are evaluated with mAP.

We use the final model weights from ImageNet-1K pre-training as network initializations and fine-tune Mask R-CNN [21] and Cascade Mask R-CNN [4] on

the COCO dataset. Following default settings in MMDetection, we use the 1× schedule (i.e.,12 epochs).

Table 13 shows object detection and instance segmentation results comparing ResNet18 and the proposed PDC-ResNet18. As we can see from the results, the proposed PDC-ResNet18 achieves an obvious better performance than the baseline ResNet18 in terms of the box and mask AP, confirming the effectiveness of the proposed polynomial learning scheme.

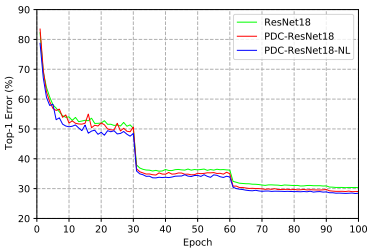

Fig. 5: Top-1 validation error on ImageNet with proposed PDC and NL methods throughout the training.