Mitigating the Effect of Incidental Correlations on Part-based Learning (Supplementary Material)

Anonymous Author(s)
Affiliation
Address
email

Supplementary Material

1 Appendix

1.1 Why the term "incidental correlations" for image background?

The concept of "incidental correlations" is derived from the notion of incidental endogeneity [3], which describes unintentional but genuine correlations between variables. In the context of our study, image backgrounds are not considered spurious because they offer contextual information that aids in decision-making. Therefore, the relationship between image backgrounds and classification is not anti-causal, as would be true if the backgrounds were spurious. We argue that the imbalance of specific image backgrounds in the training data is the primary factor contributing to the introduction of incidental correlations.

1.2 Training and inference details for pertaining and fine-tuning DPViT

Training Details. Our approach involves pre-training the Vision Transformer backbone and projection head using the same method described in the iBOT paper [8]. We mostly keep the hyper-parameter settings unchanged without tuning. By default, we use the Vit-Small architecture, which consists of 21 million parameters. The patch size is set to 16 as our default configuration. The student and teacher networks have a shared projection head for the [cls] token output. The projection heads for both networks have an output dimension of 8192. We adopt a linear warm-up strategy for the learning rate over 10 epochs, starting from a base value of 5e-4, and then decaying it to 1e-5 using a cosine schedule. Similarly, the weight decay is decayed using a cosine schedule from 0.04 to 0.4. We employ a multi-crop strategy to improve performance with 2 global crops (224×224) and 10 local crops (96×96). The scale ranges for global and local crops are (0.4, 1.0) and (0.05, 0.4), respectively. Following [8], we use only the local crops for self-distillation with global crops from the same image. Additionally, we apply blockwise masking to the global crops inputted into the student network. The masking ratio is uniformly sampled from [0, 1, 0.5] with a probability of 0.5, and with a probability of 0.5, it is set to 0. Our batch size is 480, with a batch size per GPU of 120. DPViT is pre-trained for 500 epochs for the given training set for all the datasets.

We use the value of $\lambda_{cls} = 1, \lambda_s = 0.5, \lambda_o = 0.5$ for all the datasets. In the case of ImageNet-9, we incorporate the class labels by incorporating a logit head onto the projection heads. This allows us to calculate the cross-entropy loss based on the provided class labels. The explicit utilization of class labels is necessary for the ImageNet-9 dataset because the evaluation involves straightforward classification rather than few-shot learning.

Fine-tuning Details. Once the pretraining stage is completed, we proceed to train the model using the supervised contrastive loss, which involves distilling knowledge from [cls] tokens across different views of images (referred to as $\mathcal{L}_{cls}$ in Equation 9 of the main draft). The fine-tuning process is conducted for 50 epochs using the same training data in the given dataset. We maintain the same set of hyperparameters used in the initial pretraining stage without additional tuning.

We use the value of $\lambda_{cls}^{inv} = 1$, and $\lambda_p^{inv} = 0.5$ for all the datasets.

Inference Details. For inference purposes, we utilized a feature representation obtained by the [cls] token of the teacher network. We also found concatenating the weighted average pooling of the generated patches with the [cls] token useful in a few-shot evaluation. The weights for the weighted average pooling are determined by taking the average of the attention values of the [cls] token across all heads of the final attention layer.

In the case of ImageNet-9, the logit head is used to infer the class label for the given sample in the test set.

1.3 Details regarding the multi-head attention modules

The design of our attention layers draws inspiration from the standard self-attention mechanism, commonly known as **qkv** self-attention (SA) [2]. In our implementation, we calculate a weighted sum over all values **v** in the input sequence **z**, where **z** has dimensions of $\mathbb{R}^{N \times D}$. The attention weights A_{ij} are determined based on the pairwise similarity between two elements of the sequence and their corresponding query $\mathbf{q}^i$ and key $\mathbf{k}^j$ representations.

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z}\mathbf{U}_{qkv} \qquad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}, \tag{1}$$

$$A = softmax\left(\mathbf{q}\mathbf{k}^\top / \sqrt{D_h}\right) \qquad A \in \mathbb{R}^{N \times N}, \tag{2}$$

$$SA(\mathbf{z}) = A\mathbf{v}\,. \tag{3}$$

Multihead self-attention (MSA) is an expansion of the self-attention mechanism, where we perform k parallel self-attention operations, known as "heads," and then combine their outputs through concatenation. In order to maintain consistent computation and the number of parameters when adjusting the value of k, the dimension D_h (as defined in Equation 1) is typically set to D/k.

$$MSA(\mathbf{z}) = [SA_1(z); SA_2(z); \cdots; SA_k(z)]\,\mathbf{U}_{msa} \qquad \mathbf{U}_{msa} \in \mathbb{R}^{k \cdot D_h \times D} \tag{4}$$

1.4 Details regarding the power iterative method to compute spectral norm

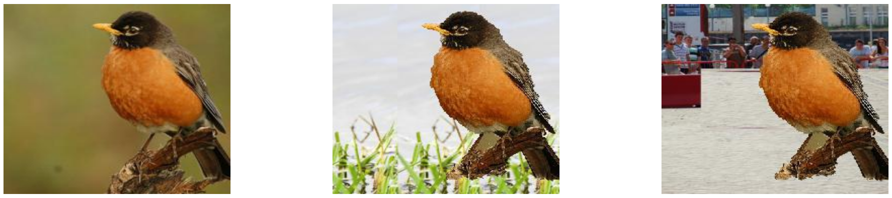
We follow the power iterative method described in [1] to compute the spectral norm for $(\mathbf{P}^T\mathbf{P} - \mathbf{I})$. Starting with a randomly initialized $v \in \mathbb{R}^n$, we iteratively perform the following procedure a small number of times (2 times by default) :

$$u \leftarrow (\mathbf{P}^T\mathbf{P} - \mathbf{I})v, v \leftarrow (\mathbf{P}^T\mathbf{P} - \mathbf{I})u, \sigma(\mathbf{P}^T\mathbf{P} - \mathbf{I}) \leftarrow \frac{||v||}{||u||}. \tag{5}$$

The power iterative method reduces computational cost from $\mathcal{O}(n^3)$ to $\mathcal{O}(mn^2)$, which is practically much faster when used with our training procedure.

Method	**IN-9L** ↑	**Original** ↑	**M-SAME** ↑	**M-RAND** ↑	**BG-GAP** ↓
ResNet-50 [6]	94.6	96.3	89.9	75.6	14.3
WRN-50×2 [6]	95.2	97.2	90.6	78.0	12.6
ConstNet [7]	90.6	92.7	86.1	69.2	17.1
ViT-S pretrained [2]	82.5	84.9	72.2	50.3	21.9
ConceptTransformer [5]	84.7	85.5	73.1	51.5	21.6
Ours - DPViT	**96.9**	**98.5**	**93.4**	**87.5**	**5.9**

Table 1: Performance evaluation on domain shift of varying background and common data corruptions on ImageNet-9. Evaluation metric is Accuracy %.



(a) Original (b) MIXED-SAME (c) MIXED-RAND

Figure 1: Visualizing the test splits from ImageNet-9 dataset.

1.5 Comparing ViT-S [2] and Concept Transformer (CT) [5] on ImageNet-9

In addition to the findings presented in Section 5.4 (Table 2 in the main draft), we conducted a comparison with vanilla ViT-S pretrained on Imagenet and ConceptTransformers (CT) as well. CT, as described in the study by [5], has a notable limitation in that it relies on attribute supervision for part localization information. This restriction restricts the applicability of CT in scenarios where attribute information is absent, such as in the case of ImageNet-9. To train CT without attributes, we utilized the code provided by the authors and deactivated the attribute loss, allowing CT to be trained without relying on the attribute information [1]. This adjustment significantly decreases the performance of CT but enables a fair comparison with other methods on ImageNet-9. It is worth noting that CT employs the ViT-S backbone pretrained on ImageNet as its default architecture. Moreover, we train ConstNet [7] using the source code provided by the authors [2].

As indicated in Table 1, DPViT demonstrates superior performance compared to both ViT-S pretrained on ImageNet and CT, exhibiting a clear advantage. CT can be seen as a pretrained ViT-S model with the inclusion of part dictionaries, but it experiences a noticeable drop in performance when confronted with the presence of incidental correlations in the image backgrounds (as observed in the low **M-SAME** and **M-RAND** performance in Table 1). This demonstrates that the part learners in general cannot effectively deal with the incidental correlations of backgrounds and are susceptible to varying backgrounds.

1.6 Ablation study with different values of K and n_f on MiniImageNet

In this analysis, we investigate the impact of varying the number of parts, denoted as K, on the MiniImageNet dataset. Specifically, we explore the effects of altering the number of foreground parts, represented by n_f, as well as the number of background vectors, which can be calculated as $K - n_f$. Table 2 presents the obtained results, demonstrating the influence of different values of K, n_f, and n_b on parts, foreground parts, and background parts, respectively.

Our findings indicate that maintaining $K = 64$ and selecting $n_f = 2K/3$ yields the highest performance. When employing a significantly lower number of part vectors, the model's capacity becomes insufficient, leading to performance degradation. Conversely, employing a larger value of K results in increased computational complexity associated with distance maps, subsequently leading to lower performance.

[1] ConceptTransformer [5] - https://github.com/IBM/concept_transformer

[2] ConstNet [7] - https://github.com/mlpc-ucsd/ConstellationNet

	K=32		K=64		K=96		K=128	
Foreground parts	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
$n_f = K/2$	$72.2_{\pm 0.2}$	$87.8_{\pm 0.4}$	$72.9_{\pm 0.5}$	$88.1_{\pm 0.4}$	$72.1_{\pm 0.2}$	$88.1_{\pm 0.4}$	$72.1_{\pm 0.5}$	$87.1_{\pm 0.5}$
$n_f = 2K/3$	$72.2_{\pm 0.2}$	$88.1_{\pm 0.4}$	$73.8_{\pm 0.5}$	$89.3_{\pm 0.4}$	$73.1_{\pm 0.2}$	$88.1_{\pm 0.4}$	$72.2_{\pm 0.5}$	$87.4_{\pm 0.5}$
$n_f = 4K/3$	$72.3_{\pm 0.2}$	$88.4_{\pm 0.4}$	$73.4_{\pm 0.5}$	$88.5_{\pm 0.4}$	$73.2_{\pm 0.2}$	$87.9_{\pm 0.4}$	$72.5_{\pm 0.5}$	$87.8_{\pm 0.5}$

Table 2: Ablation of varying the number of foreground-background vectors, along with part-vectors used. We show the results on the miniImageNet dataset.

Setting	1-shot ↑	5-shot ↑	$\|\|\mathbf{P}\|\|_1$ ↓	$\|\|\mathbf{PP}^T - \mathbf{I}\|\|_1$ ↓
Shared	73.6	89.6	0.4	0.5
Unshared	73.8	89.8	0.3	0.5

Table 3: **Siamese DPViT**. Sharing MSA and MCA layers and evaluation on MiniImageNet.

Method	1-shot ↑	5-shot ↑
SMKD [4]	60.93	80.38
DPViT	62.81	83.25

Table 4: Few-shot performance after 1^{st} stage pretrain phase on MiniImageNet.

1.7 Computational complexity of DPViT

Adding part-dictionaries to MCA layers slightly increases the trainable parameters from $21M$ (ViT-S) to $25M$ (DPViT). It is also possible to share the attention layers, analogous to the Siamese networks, for MSA and MCA, which keeps the number of trainable parameters to $21M$. DPViT results in a similar performance in terms of few-shot accuracy when the attention layers are shared, as shown in Table 3.

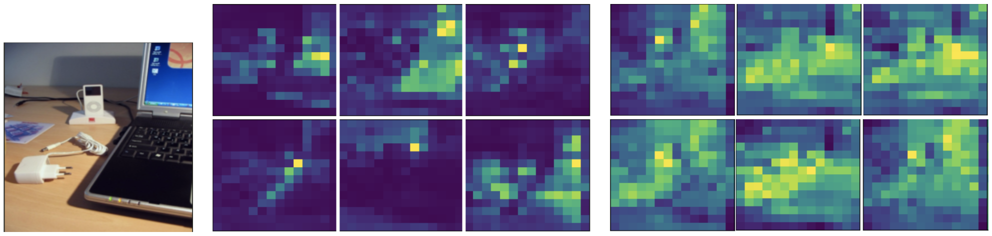
1.8 Stage-1 pertaining comparison with SMKD [4]

Table 4 showcases the few-shot evaluation results of DPViT on the MiniImageNet dataset. In addition, we compare the performance of DPViT with the first-stage performance of SMKD [4]. It is worth noting that both DPViT and SMKD utilize the iBOT [8] pretraining strategy. However, incorporating part-dictionaries, MSA, and MCA layers in DPViT's pretraining phase contributes to its superior performance compared to SMKD.

1.9 Studying complementary properties of MSA and MCA

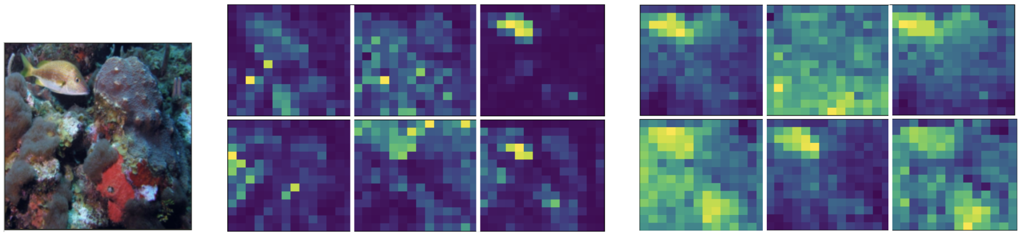
Based on Section 5.1 in the main draft, our study focuses on examining the complementary characteristics of MSA and MCA. MSA is designed to be effective for images containing a small number of objects, but it struggles to capture the spatial relationships among multiple objects. In contrast, MCA layers utilize distance maps to learn spatial relationships and prioritize objects without considering their specific classes. In simpler terms, MSA may overlook certain objects that are not crucial for classification, while MCA emphasizes learning spatially similar objects.

Additionally, we present the visualization of attention heads in Figure 2, 3, and 5. The MSA heads excel at identifying objects for classification but may overlook relevant objects with significant spatial context, such as the "charger" in Figure 2 and the "garbage box" in Figure 5. On the other hand, the MCA layers perform well in scenarios involving multiple objects (Figure 2 and 5), but struggle when spatially similar objects are present, as seen with the confusion between the "red grass" and the "fish" in Figure 3.



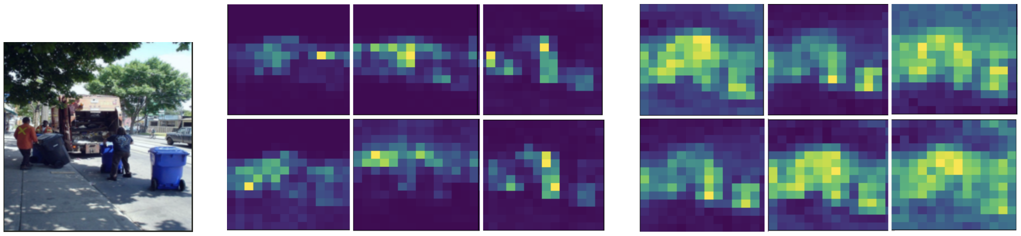
(a) Original (b) Attention heads of MSA (c) Attention heads of MCA

Figure 2: Visualizing the attention heads for MSA and MCA.



(a) Original (b) Attention heads of MSA (c) Attention heads of MCA

Figure 3: Visualizing the attention heads for MSA and MCA.



(a) Original (b) Attention heads of MSA (c) Attention heads of MCA

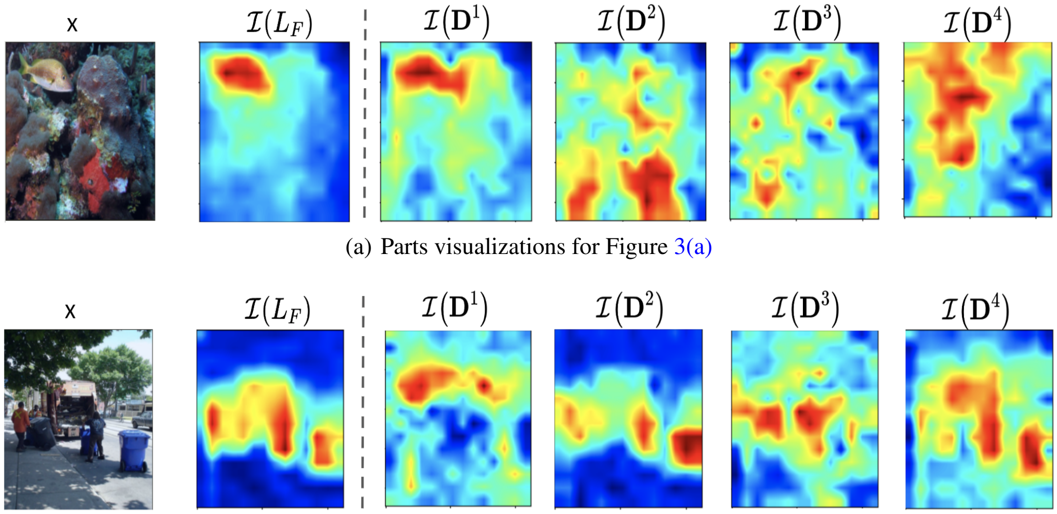
Figure 4: Visualizing the attention heads for MSA and MCA.

1.10 Visualization foreground parts

We present additional part visualizations for Figure 3(a) and 4(a). These are shown in Figure 5(a) and 5(b).

1.11 Qualitative comparison of extracted patches with ConstNet [7]

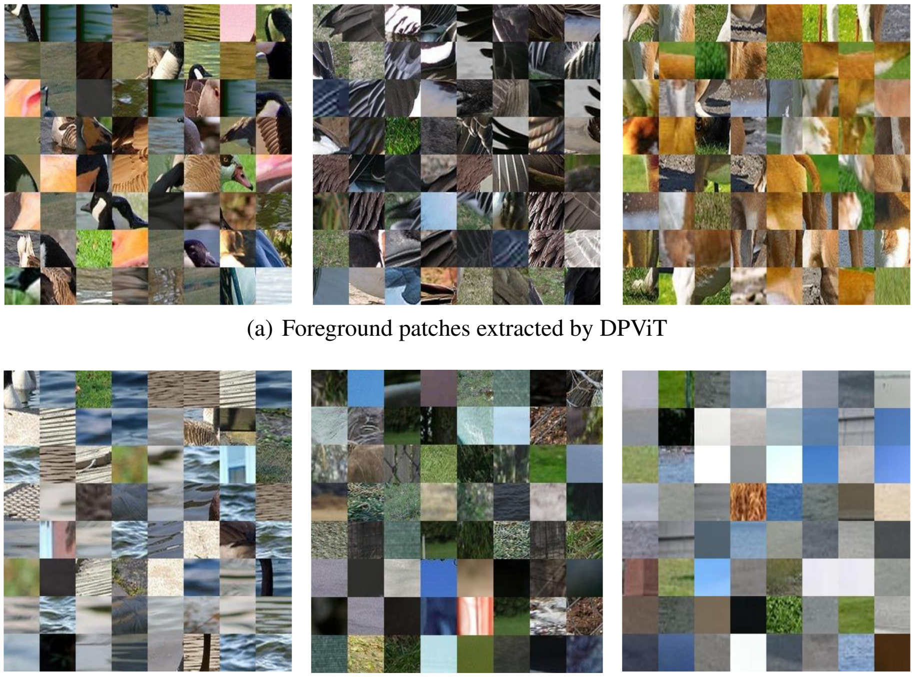
In order to showcase the acquired parts of DPViT and ConstNet, we provide visualizations in Figure 6 and 7. This is achieved by selecting the nearest patches to the parts. Figure 6 illustrates the separation



(a) Parts visualizations for Figure 3(a)

(b) Parts visualizations for Figure 4(a)

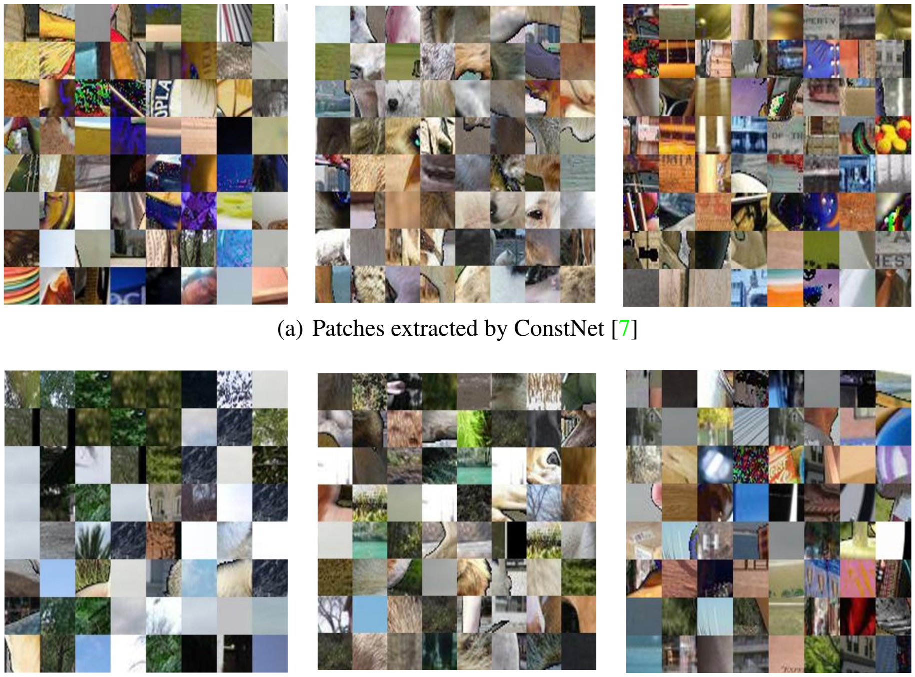
Figure 5: Visualizing foreground parts learned by DPViT.



(a) Foreground patches extracted by DPViT

(b) Background patches extracted by DPViT

Figure 6: Visualizing foreground and background patches extracted by DPViT around a random foreground and background part for images from the validation set of MiniImageNet.



(a) Patches extracted by ConstNet [7]

(b) Patches extracted by ConstNet [7]

Figure 7: Visualizing patches extracted by ConstNet [7] around a random parts for images from the validation set of MiniImageNet.

133 of foreground and background concepts accomplished by our model, whereas Figure 7 exhibits the
134 patches surrounding the learned parts from the ConstNet model.

While DPViT learns to disentangle the foreground patches from the backgrounds, the patches extracted by ConstNet suffer from the entanglement caused due to incidental correlations of backgrounds.

References

[1] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

[3] Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014.

[4] Han Lin, Guangxing Han, Jiawei Ma, Shiyuan Huang, Xudong Lin, and Shih-Fu Chang. Supervised masked knowledge distillation for few-shot transformers. *arXiv preprint arXiv:2303.15466*, 2023.

[5] Mattia Rigotti, Christoph Miksovic, Ioana Giurgiu, Thomas Gschwind, and Paolo Scotton. Attention-based interpretability with concept transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

[6] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *International Conference on Learning Representations (ICLR)*, 2020.

[7] Weijian Xu, Yifan xu, Huaijin Wang, and Zhuowen Tu. Attentional constellation nets for few-shot learning. In *International Conference on Learning Representations*, 2021.

[8] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.