Supplementary Material for Attribute Surrogates Learning and Spectral Tokens Pooling in Transformers for Few-shot Learning

A. More Ablation Results

Extended experiments are conducted to further validate our design's effectiveness.

Do class and patch surrogates need different semantic spaces? In our proposed methods, we expect class surrogates and patch surrogates to reside in different semantic spaces to complete each other by solely feeding the class output into a MLP layer. Experimental results show that without this re-projecting procedure, performance drops significantly by 29.5%. It suggests that when constrained within the same feature space, supervision on the class surrogate and patch surrogate might lead to severe confusion that hinders efficient learning.

Are local views harmful to surrogate learning? For every input image, our training framework produces several local and global views for self-supervision. When imposing additional surrogate-level supervision, we only attend to these global views for their better information perseverance. Experimental results show that with both global and local views supervised, performance drops by 1.9% , corresponding to the second line with views global+local in Table 1. It suggests that local views tend to introduce noise for surrogate learning, which is natural considering the non-neglected object truncation issues in local views.

Why not use patch loss in latter stages? In our design, we choose to let latter transformers inherit $[class]$ token from the previous stage for initialization and impose no surrogate supervision on patch levels. We conduct experiments where patch-level supervision is preserved in latter transformer sets. As shown in Tab. 2, with patch-level supervision, results barely improve over the basis of stage 1 and are inferior to our design with class-level supervision only. We suggest that the reason lies in that at later stages, complementary characteristics of $[cls]$ surrogates are key to further performance improvement. With patch loss added, the network will lose its focus in $[cls]$ surrogate learning and the intertwined training will lose previous $[cls]$ surrogates' good properties.

Do higher resolutions always lead to better performance? Our proposed method uses the image with resolution 224×224 as input, which is different from existing SOTA methods. To prove that our performance elevation is not a resolution gain, we conduct experiments with two SOTA methods in 224×224 settings in Table 7 of our paper. Results show that higher resolutions lead to performance drop because of the increased tendency to overfitting for these methods. For supplementary, results with Meta-baseline [10] are listed in 3. These results show that our method's improvement is not introduced by higher resolution and possesses stronger generalization ability.

B. Qualitative Results

To further validate that our method can retrieve meaningful semantic representations with small datasets, we visualize self-attention maps associated with $[cls]$ token and show results from our retrained DINO baseline and DINO +CE(with class-level cross entropy supervision) in Fig. 1 for a fair comparison. Results show that our method focuses more on foreground information than the other two methods. In addition, we listed more visual results of two consecutive spectral tokens pooling procedures in Fig. 2.

C. Comparison on Parameters and Computational Cost

Learnable Parameters Comparision. In Table 4, we can find that more learnable parameters don't lead to better performance directly. The backbone of METAQDA $_{\text{ICCV21}}$ has much more parameters than IE $_{\text{CVPR21}}$ and HCTransformers, but

Method	f_c(x)	P(x)	views	1-shot	5-shot
DINO	✓	-	global	41.79 ± 0.17	56.27 ± 0.15
DINO	-	✓	global+local	69.37 ± 0.16	82.99 ± 0.11
Ours	-	✓	global	$\mathbf{71.27 \pm 0.17}$	$\mathbf{84.68 \pm 0.10}$

$f_c(x)$: the encoded $[cls]$ token, $P(x)$: the $[cls]$ token after projection head

Table 1. Results of the first student transformer trained with different surrogate space and image view settings on *mini*Imagenet. All models are based on the DINO baseline. Two choices for class surrogate are tested : $f_c(x)$, the $[cls]$ token extracted from ViT encoder, and $P(x)$, projection of $[cls]$ token via classification head. To explore impacts of local views for surrogate-level supervision, experiment with all views supervised is conducted for fair comparison.

Loss	stage1	stage2	stage3
DINO+CLS+PTH	71.27 ± 0.17	71.25 ± 0.17	71.19 ± 0.17
DINO+CLS	-	$\mathbf{74.74 \pm 0.17}$	72.66 ± 0.18

DINO+CLS: combination of the class surrogate loss and the DINO loss.
DINO+CLS+PTH: full combination of the class surrogate loss, the patch surrogate loss and the DINO loss.

Table 2. Results of supervising learning process for the latter two transformer sets with or without patch surrogates.

Method	resolution	*mini*Imagenet		*tiered*Imagenet	
		1-shot	5-shot	1-shot	5-shot
Meta-baseline [10]	84^2	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.74 ± 0.18
Meta-baseline [10]	224^2	67.20 ± 0.23	81.19 ± 0.16	70.28 ± 0.27	82.24 ± 0.20
Ours-Cosine	224^2	74.74 ± 0.17	85.66 ± 0.10	79.67 ± 0.20	89.27 ± 0.13

Table 3. Comparison with the state-of-the-art 5-way 1-shot and 5-way 5-shot performance with 95% confidence intervals on different resolutions on ***mini*Imagenet** and ***tiered*Imagenet**.

get inferior performances. Compared with the ViT-S backbone and DINO $_{\text{ICCV21}}$, HCTransformers get impressive improvements. It indicates that the proposed attribute surrogates learning and spectral tokens pooling is very important to utilize the strong learning abilities of transformers. Although we have more parameters than IE $_{\text{CVPR21}}$ with the ResNet-12 backbone, we argue that our contribution is improving the data efficiency for transformers, and thus making them suitable for few-shot learning.

Method	Backbone	Params	1-shot	5-shot
IE $_{\text{CVPR21}}$	ResNet-12	12.4M	$69.28_{\pm 0.80}$	$85.16_{\pm 0.52}$
METAQDA $_{\text{ICCV21}}$	WRN-28-10	36.5M	$67.38_{\pm 0.55}$	$84.27_{\pm 0.75}$
Cosine Distance	ResNet-50	23M	$59.28_{\pm 0.20}$	$72.68_{\pm 0.16}$
Cosine Distance	ViT-S	21M	$52.92_{\pm 0.17}$	$65.04_{\pm 0.14}$
DINO $_{\text{ICCV21}}$(baseline)	ViT-S	21M	$61.57_{\pm 0.16}$	$75.51_{\pm 0.12}$
HCTransformers 1	ViT-S	21M	$71.27_{\pm 0.17}$	$\mathbf{85.66}_{\pm 0.10}$
HCTransformers 2	ViT-S	21M	$\mathbf{74.74}_{\pm 0.17}$	$\mathbf{89.19}_{\pm 0.13}$
HCTransformers 3	ViT-S	21M	$\mathbf{72.66}_{\pm 0.17}$	$\mathbf{85.66}_{\pm 0.10}$

Table 4. Comparison of state-of-the-art algorithms with different backbones on *mini*Imagenet.

Computational Cost of Spectral Tokens Pooling. In Table 5, we list the training time cost of different modules in HCTransformers. It can be found that the spectral tokens pooling is relatively slow in our whole pipeline. But when compared with the first training stage, the time spent is still affordable because it needs only several epochs to train the second and third sets of transformers.

	Stage 1(400 epoch)	Stage 2 (2 epoch)		Stage 3 (2 epoch)	
	ViT	Pooling	ViT	Pooling	ViT
Time	21.1h	0.33 h	0.25 h	0.12 h	0.09 h

Table 5. The amount of training time spent at each stage on 8 Nvidia RTX 3090 GPUs on *mini*ImageNet.

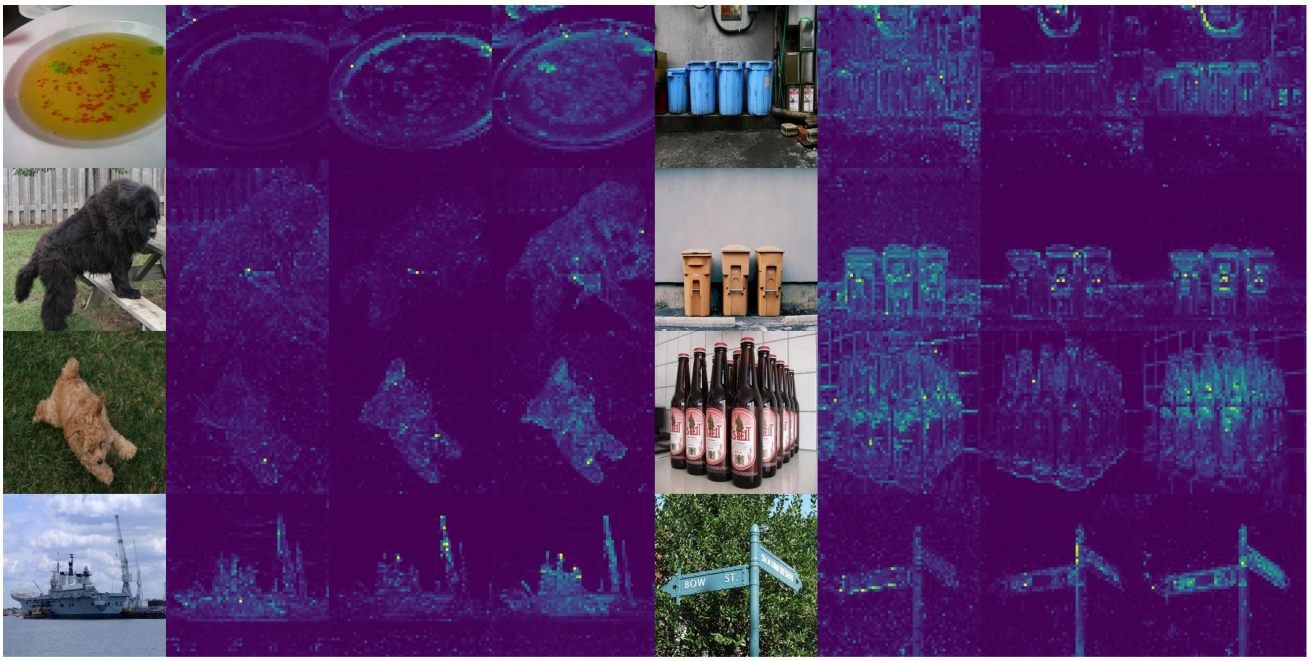

Figure 1. Visualization of Self-attention maps with the $[cls]$ token in final layers by DINO baseline, DINO + CE(with class-level supervision) and our proposed method.

Figure 2. More visualization of tokens pooling process. After spectral tokens pooling operations, adjacent tokens with similar semantics are clustered into one. (c) and (d) shows that our clustering results are well consistent with the image's basic structure. The pixel colors in the same cluster are averaged.