Task-Adaptive Saliency Guidance for Exemplar-free Class Incremental Learning –Supplementary Material–

Xialei Liu [2,1,*] Jiang-Tian Zhai [1, *] Andrew D. Bagdanov [3] Ke Li [4] Ming-Ming Cheng [2,1, †]
[1] VCIP, CS, Nankai University [2]NKIARI, Shenzhen Futian [3] MICC, University of Florence [4] Tencent Youtu Lab
{xialei,cmm}@nankai.edu.cn, {jtzhai30,tristanli.sh}@gmail.com, andrew.bagdanov@unifi.it

Method	5 Tasks	10 Tasks	20 Tasks
baseline (SSRE)	40.2	40.0	39.3
CAM	41.2	40.7	40.4
SmoothGrad	42.1	41.0	40.4
Grad-CAM	44.1	43.9	43.5

Table 1. Ablation on methods for generating saliency maps on Tiny-Imagenet.

A. Further Ablation Studies

Ablation on saliency methods. To show the generalization of TASS, we use several methods to compute saliency maps and report the results in Table 1. Grad-CAM performs the best, although other methods yield performance gains, demonstrating the effectiveness of TASS.

Ablation on low-level target maps. In the manuscript we use CSNet [1] to compute all the pre-trained saliency and boundary maps because it is very lightweight. Compared to our main model, the pre-trained model has fewer than 1% parameters and requires 1.5% of the FLOPs (as shown in Table 2). Note that we compute all low-level maps offline before new tasks, and so the extra FLOPs should be amortized over the number of epochs. Therefore, the additional FLOPs required by the low-level model is only about 0.015% of the main model, which is negligible in practice.

To show the effectiveness of TASS, we perform an ablation on the low-level maps. We replace them with the Grad-CAM generated from a ResNet-152 network. To avoid information leakage, ResNet-152 was trained from scratch. Before each new task, we first train it only on task data and use the Grad-CAM output to supervise saliency in our incremental model. From Table 3 we see that TASS still outperforms other methods. Moreover, TASS is applicable to other models for generating saliency maps, (e.g. DFI [4] or PoolNet [3]) with more parameters, and produces even better performance with larger networks.

Model	Parameter(M)	FLOPS(G)
Ours	17.9	0.78
Pre-trained salient model	0.0941	0.012

Table 2. Parameters and FLOPs of the pre-trained salient model. FLOPs are computed using $3 \times 32 \times 32$ images.

Low-level source (Method)	Accuracy (%)
PASS	39.3
SSRE	40.0
ResNet152 (Ours)	42.1
CSNet (Ours)	43.9
PoolNet (Ours)	44.2
DFI (Ours)	44.4

Table 3. Ablation on low-level saliency maps on Tiny-ImageNet with 10 tasks.

Ablation on method architecture and salient model pretraining. We select PASS [7] as our baseline method to apply TASS to (as shown in Table 4 and Table 5). Experiments in these two tables are conducted on ImageNet-Subset with 5 tasks. Since some methods use ImageNet pretrained weights for better saliency map estimation, we train CSNet [1] from scratch on the dataset (with and without pretraining) for salient object detection [2, 5, 6]. This allows us to verify that no information leakage happens due to pretraining the saliency network on ImageNet. The low-level network without pretraining works almost as well as pretraining the saliency network on ImageNet. We also compare the number of parameters of different methods in Table 4. This shows that adding network capacity for PASS from ResNet-18 to ResNet-32 with more parameters only improves the performance marginally. Ours with ResNet-

[*] The first two authors contribute equally.
[†] Corresponding author

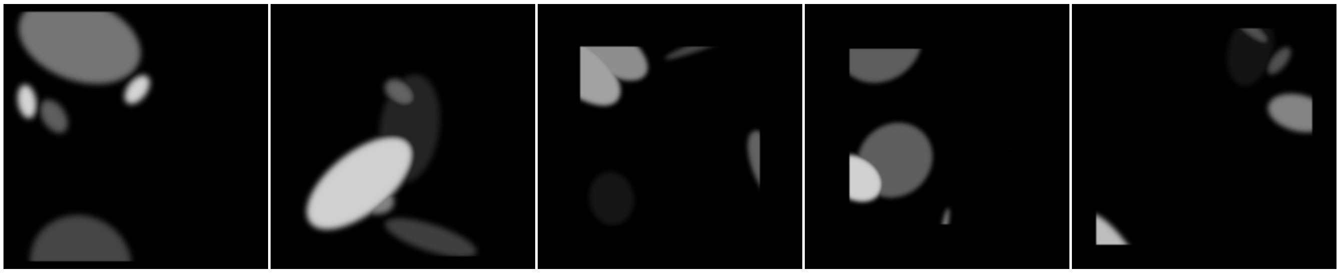

Figure 1. Visualization of some generated saliency noise maps.

Method	Parameter (M)	Accuracy (%)
PASS-Res18	14.5	50.4
PASS-Res32	21.7	51.2
SSRE-Res18	19.4	58.7
Ours-Res18	17.9	61.5

Table 4. Comparison of different method network architectures. Method-Res18 denotes applying Method with ResNet18 as its backbone.

Method	Accuracy (%)
No pretraining	61.5
Pretrained salient detection model	62.0

Table 5. Ablation on salient detection network pretraining.

18 based on PASS achieves a significant gain surpassing SSRE which has more parameters.

Our approach in non-DFCIL scenarios. We apply our saliency supervision in a non-DFCIL scenario using PASS in Table 6 by including 20 exemplars per class. TASS boosts performance significantly here as well.

Method	buffer size	Acc (%)
PASS	20	52.36
PASS+TASS	20	**55.75**

Table 6. TASS on a non-DFCIL scenario.

Hyper-parameters of multiple losses. In Eq. 5 we weight all loss terms equally. As suggested, we explore more options in Table 7. Tuning further improves the performance slightly, but we stick with $\lambda_{\text{CIL}} = \lambda_{\text{lm}} = \lambda_{\text{dbs}} = 1.0$ for convenience. $\sqrt{N}$ in Eq. 2, where N is the number of pixels, is used to normalize the L2 distance.

All loss permutations ablation. We give all possible combinations of all three loss terms in Table 8. These results show that each component contributes to the final performance and that a combination of them performs best.

λ_{CIL}	λ_{lm}	λ_{dbs}	Acc(%)
1	1	1	55.01
0.1	1	1	**55.27**
1	0.1	1	54.22
1	1	0.1	54.31

Table 7. Hyper-parameters of multiple losses for SSRE+TASS.

Method		L	D	S	LD	LS	DS	LDS
PASS	49.0	51.2	50.6	51.4	53.0	52.6	53.7	54.5
SSRE	55.0	56.2	55.8	56.7	57.3	57.0	57.6	57.9

Table 8. LDS represent Low-level multi-task supervision, Dilated boundary supervision, and Saliency noise injection, respectively.

F & Acc (%)	50	30	10	0
PASS	49.03±0.9	46.78±0.9	44.65±1.0	40.27±1.0
PASS+TASS	54.45±0.4	51.22±0.5	48.58±0.5	44.30±0.5

Table 9. Ablation on F with $T = 10$.

Class division in experimental protocol. We follow conventional experimental setups from previous works like PASS and SSRE to divide the classes of the dataset as $F + C \times T$ with $F = 50$. As suggested, we evaluate different options for F in Table 9. TASS shows consistent gain compared to the baseline under all settings.

B. More Visualizaions on TASS

Saliency Noise. For each ellipse there are 6 dimensions: the center coordinate (x, y), the rotation angle α, the mask weight w, and the major and minor axes (a, b). x, y, α and w are sampled from a uniform distribution over ranges: $x \in [0, H)$, $y \in [0, W)$, $\alpha \in [0, 2\pi)$, $w \in [0, 1]$. H and W denote the height and width of input images. To generate ellipses of appropriate size, we draw the major and minor axes from a Gaussian distribution with $\mu_a = \max(H, W)/2$, $\sigma_a = \max(H, W)/6$, $\mu_b = \min(H, W)/2$, $\sigma_b = \min(H, W)/6$. The sampled a, b is clipped to$[0, max(H, W)/2]$ and $[0, min(H, W)/2]$, respectively. For each ellipse, we create a saliency map S_i. We repeat this random generation process 3-5 times and apply an element-wise max operation on the S_i to obtain a single saliency map S. Then we crop and resize S to

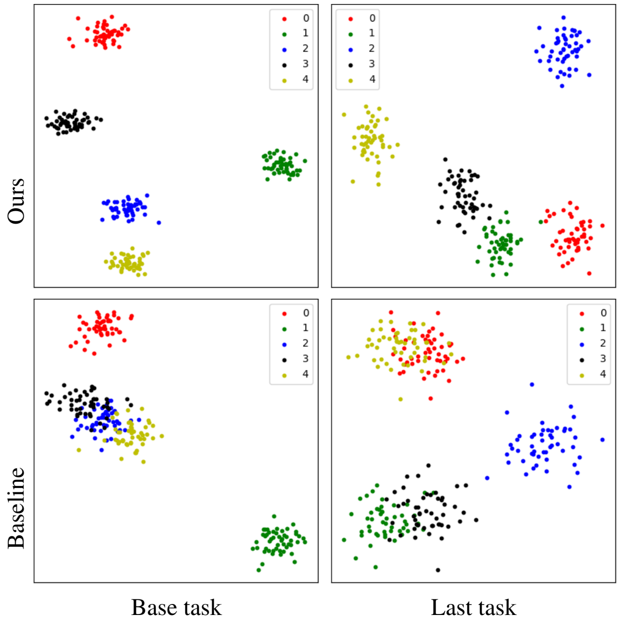

Figure 2. Visualization of the embedding $F_\phi(x)$ with and without TASS. Compared to the baseline, our method preserves more discriminative representations.

the original image size, with crop size sampled from a uniform distribution in $[\min(H, W)/2, \min(H, W)]$, introducing center-aware saliency noise to the network for training. Finally, we apply a Gaussian blur on S to better simulate a realistic saliency map. The kernel size for Gaussian blurring is the closest odd integer to $\min(H, W)/20$. For each encoder feature map, 10% of randomly selected channels are directly masked with S, where each selected channel will have an independent S. We visualize several generated samples in Figure 1.

Embedding Visualization. Since our method helps the model focus on the foreground, more class-specific pixels contribute to the embedding. Thus embeddints are more discriminative and contain less distracting background information. In Figure 2 we use t-SNE to visualize embeddings of five initial classes after learning the base and last task in the 10-task setting on ImageNet-Subset. At the base task, both Baseline (SSRE) and Ours (SSRE+TASS) perform well. After the last task, it is clear that TASS helps maintain discriminative features between tasks while the Baseline has overlapping embeddings.

References

[1] Ming-Ming Cheng, Shanghua Gao, Ali Borji, Yong-Qiang Tan, Zheng Lin, and Meng Wang. A highly efficient model to study the semantics of salient object detection. *IEEE TPAMI*, 2021. 1

[2] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 1

[3] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, 2019. 1

[4] Jiang-Jiang Liu, Qibin Hou, and Ming-Ming Cheng. Dynamic feature integration for simultaneous detection of salient object, edge and skeleton. *IEEE TIP*, 2020. 1

[5] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, 2013. 1

[6] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 1

[7] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, 2021. 1