Unsupervised Dense Prediction using Differentiable Normalized Cuts (Supplementary Material)

Yanbin Liu[1] and Stephen Gould[2]

[1] Auckland University of Technology
[2] Australian National University
yanbin.liu@aut.ac.nz, stephen.gould@anu.edu.au

1 Technical Details

Training and Loss details. We first describe a symmetric version of the mask consistency loss, denoted as $\mathcal{L}'_{mask}$. In this case, the features are obtained as: $\boldsymbol{f}'_1 = g(\Phi_\theta(\mathbf{I}_1)), \boldsymbol{f}'_2 = q(g(\Phi_\theta(\mathbf{I}_2)))$, which swaps the features before Projector in Fig. 2. And $\boldsymbol{f}_1^{\text{ovlp}'}, \boldsymbol{f}_2^{\text{ovlp}'} = \text{RoIAlign}(\boldsymbol{f}'_1, \boldsymbol{f}'_2, t_1, t_2)$. Then $\boldsymbol{m}_1^{\text{ovlp}'}, \boldsymbol{m}_2^{\text{ovlp}'}$ can be computed by NCut [5]. Finally,

$$\mathcal{L}'_{mask} = \mathcal{L}_{bce}(\texttt{stop\_grad}(\boldsymbol{m}_1^{\text{ovlp}'}), \boldsymbol{m}_2^{\text{ovlp}'}). \tag{1}$$

We only compute losses for examples that have an $\text{IoU}(\boldsymbol{m}_1^{\text{ovlp}'}, \boldsymbol{m}_2^{\text{ovlp}'}) > 0.5$.

Given $\boldsymbol{m}_2^{\text{ovlp}'}$, we also enforce the structural regularization on it, similar to Eqn. 11. Moreover, the symmetric version of the horizontal and vertical masks (Fig. 3) is utilized for regularization.

For training, we adopt the Layer-wise Adaptive Rate Scaling (LARS) on top of an SGD optimizer with a momentum of 0.9.

Structure of the Projector and Predictor. Both the Projector and Predictor are two-layer 2D ConvNets. The structure of the Projector is as follows *Conv2D(384, 384) – BN – GELU – Conv2D(384, 256)*. The structure of the Predictor is as follows *Conv2D(256, 384) – BN – GELU – Conv2D(384, 256)*. The number inside the parentheses indicates the input and output channels.

Prevent numerical instablity. One issue of differentiating the eigendecomposition problem is the numerical instability when two eigenvalues λ_i, λ_j get close to each other[3]. For example, the Pytorch function `torch.linalg.eigh` encounters this issue. Let $K_{ij} = \frac{1}{\lambda_i - \lambda_j}$. In this paper, we deal with the numerical issue by setting $K_{ij} = 0$ when $|\lambda_i - \lambda_j| < 10^{-9}$.

[3] `https://pytorch.org/docs/stable/generated/torch.linalg.eigh.html`

	VOC07	VOC12	COCO20k
Ours	**71.3**	**74.2**	**61.6**
w/o structural regularization	64.0	69.1	53.4
w/o projector	56.5	58.0	53.1
pixel consistency	62.0	67.1	52.2

Table 1: Ablation study of design choices for Object Discovery Task. We use the TokenCut [7] post-processing to extract bounding boxes. *CorLoc* score is reported.

α	DUT-OMRON	DUTS-TE	ECSSD
0.0	48.8	55.8	70.0
0.025	52.8	59.9	72.5
0.05	**56.4**	**61.8**	**72.9**
0.1	54.9	59.2	72.5
0.2	49.4	51.6	70.8
0.4	34.4	37.2	54.4

Table 2: Effect of the regularization factor α. We use the "Simple" post-processing: *taking* $\boldsymbol{z}_2 > \mathbf{0}$ *as foreground*. IoU is reported. The best performance is shown in **bold**.

2 More Ablation Study

Ablation on Object Discovery. In Tab. 4 of the main paper, we ablated the design choices on Saliency Detection. Here, we conducted similar ablation for the Object Discovery Task, and show the results in Tab. 1. A similar conclusion can be drawn that each of our model components (*i.e.*, structural regularization, Projector, mask-consistency loss) is important for the overall performance.

Effect of the regularizer factor α**.** We then study the effect of the structural regularization parameter α, and show the results in Tab. 2. It can be seen that too strong or too weak regularization is harmful to performance. The best results are achieved when $\alpha = 0.05$.

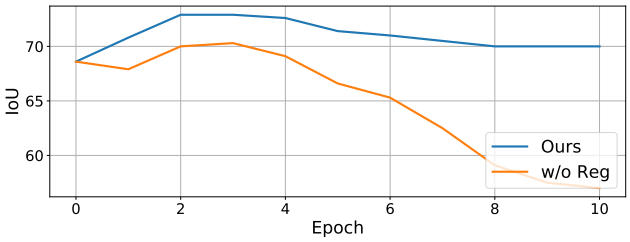

Fig. 1: IoU on ECSSD dataset with different finetuning epochs.

Method	Architecture	Forward	Backward	Total
Pseudo inverse	ViT-S/16	0.6677	4.8206	5.4883
Pseudo inverse	ViT-S/8	0.7616	8.3074	9.0690
Ours	ViT-S/16	0.6677	0.0011	0.6688
Ours	ViT-S/8	0.7616	0.0013	0.7629

Table 3: Running time (seconds) comparison of Pseudo-inverse and Our method. All forward passes adopted the same `torch.linalg.eigh`[4] function, while backward utilized either Eqn. 7 or Eqn. 8.

Effect of the training epochs. We show the IoU *w.r.t* different epochs in Fig. 1. Our method is stable and peaks at epoch two. However, without regularization (Eq. 11), the performance drops quickly after three epochs, showing the necessity of regularization.

Running time comparison of Pseudo-inverse and Our method. The original formulation of Magnus *et al.* [4] involves a Pseudo-inverse operation (Eqn. 7), which is time-consuming. We propose a more efficient formulation by utilizing eigen decomposition (Eqn. 8). To compare the running time, we perform experiments on NVIDIA A100 80GB GPU. For ViT-S/16, the batch size is 256, and for ViT-S/8, the batch size is 64. The time comparison is reported in Tab. 3. We can find that in the forward pass, the running time is the same for both methods. But *in the backward, the time cost of our method is neglected due to the simple reciprocal operation and re-use of forward matrices.* In contrast, Pseudo-inverse has a large time cost due to the complex inverse operation. Overall, our method is over 8× faster compared with the Pseudo-inverse.

Evolving pattern of the mode collapse. We show results as the training iteration increases to understand the possible collapse patterns of the dense learning framework. In the beginning, the mask is clear and well-structured. With training continuing, the mask becomes blurred and gradually loses its structure. Finally, the mask collapses to a horizontal striped pattern. With structural regularization, the mask always remains meaningful and sharp.

3 Object Discovery Generalizes Beyond DINO

In the main paper (Section 4.3 and Tab. 3), we have verified that our method can generalize to various pre-training methods (MAE [3] and MocoV3 [2]) beyond DINO and various network configurations (ResNet, ViT-S/16, ViT-S/8 and ViT-B/16), on Unsupervised Saliency Detection datasets. In order to verify the

[4] The backward pass of Pytorch is numerically unstable and has several constraints: (1) Gradients computed using the eigenvectors tensor will only be finite when A has distinct eigenvalues. (2) Furthermore, if the distance between any two eigenvalues is close to zero, the gradient will be numerically unstable. In contrast, our implementation (Eqn. 8) is numerically stable and efficient.

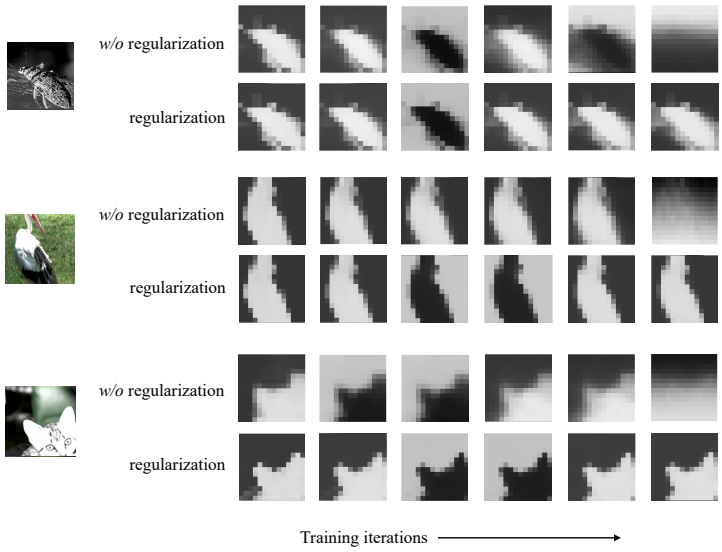

Fig. 2: Training Collapse. Without regularization, the mask gradually loses structural information and gets blurred. Finally, it collapses to a horizontal striped pattern. With our structural regularization, the mask always keeps meaningful and sharp. Note that due to the simple heuristics (*i.e.*, *taking second smallest eigenvector* $\boldsymbol{z}_2 > \mathbf{0}$ *as the foreground*), the foreground and background masks sometimes change.

generalization capability across different tasks, we do the same experiments on the Unsupervised Object Discovery datasets. The results are shown in Tab. 4. A similar conclusion can be drawn for object discovery: our method can significantly improve the performance of existing pre-trained models at a cheap cost (*i.e.*, two epochs of finetuning).

4 More Visualization Results

We show more visualization results in Fig. 3. We observed several interesting examples. In the first row, our trained model focuses more on the person, while the original DINO [1] focuses on the reflection. This results in different saliency masks in Fig. 3(e) and (f). In the second row, our model is able to provide accurate attention to tiny objects. In the third row, our model attends to the foreground person, while DINO attends to the noisy background. Note that our method also has failure cases, as shown in the last two rows. When there are multiple objects in an image, applying our model to `FOUND` [6] sometimes results in larger background saliency.

In Normalized Cuts [5] theory, the eigenvectors $\boldsymbol{z_2}$[5], $\boldsymbol{z_3}, \boldsymbol{z_4}, \boldsymbol{z_5}, \ldots$, naturally decompose an image into semantic segments, as shown in Fig. 4. We select $\boldsymbol{z_2}$ to

[5] The smallest eigenvector $\boldsymbol{z_1} = \mathbf{1}$ is trivial.

Post-process	Pre-training	Backbone	VOC07	VOC12	COCO20k	**Avg *CorLoc*↑**
Simple $z_2 > 0$	DINO [1]	ResNet50	45.4	51.2	34.9	
	Ours + DINO		52.0(6.6)	58.2(7.0)	40.4(5.5)	+**6.4**
	DINO [1]	ViT-S/16	63.2	66.2	52.7	
	Ours + DINO		67.7(4.5)	71.2(5.0)	57.4(4.7)	+**4.7**
	MoCoV3 [2]	ViT-S/16	57.7	60.3	45.8	
	Ours + MoCoV3		64.3(6.6)	68.2(7.9)	52.7(6.9)	+**7.1**
	MAE [3]	ViT-B/16	58.7	62.1	46.3	
	Ours + MAE		69.4(10.7)	72.9(10.8)	58.1(11.8)	+**11.1**
TokenCut [7]	DINO [1]	ResNet50	47.3	52.2	37.4	
	Ours + DINO		56.0(8.7)	62.3(10.1)	46.2(8.8)	+**9.2**
	DINO [1]	ViT-S/16	68.8	72.1	58.8	
	Ours + DINO		71.3(2.5)	74.2(2.1)	61.6(2.8)	+**2.5**
	MoCoV3 [2]	ViT-S/16	65.1	69.2	53.1	
	Ours + MoCoV3		67.5(2.4)	71.9(2.7)	56.8(3.7)	+**2.9**
	MAE [3]	ViT-B/16	58.1	63.8	42.4	
	Ours + MAE		66.3(8.2)	71.7(7.9)	54.6(12.2)	+**9.4**

Table 4: Object detection improvements across various pre-training methods, architectures and post-processing methods. "Simple" refers to our simple dataset-agnostic post-processing heuristics, *i.e.*, *employing the second smallest eigenvector* $\boldsymbol{z}_2 > \mathbf{0}$ *for foreground delineation.* The *CorLoc* improvements are shown in the parentheses. The average improvements are shown in the rightmost column.

obtain a mask for dense prediction. However, our model can be easily extended to include multiple eigenvectors for visual semantics.

References

1. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
2. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. in 2021 ieee. In: CVF International Conference on Computer Vision (ICCV). pp. 9620–9629 (2021)
3. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
4. Magnus, J., Neudecker, H.: Matrix differential calculus with applications in statistics and econometrics. Wiley series in probability and mathematical statistics Show all parts in this series (1988)
5. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22**(8), 888–905 (2000)
6. Siméoni, O., Sekkat, C., Puy, G., Vobeckỳ, A., Zablocki, É., Pérez, P.: Unsupervised object localization: Observing the background to discover objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3176–3186 (2023)
7. Wang, Y., Shen, X., Hu, S.X., Yuan, Y., Crowley, J.L., Vaufreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14543–14553 (2022)

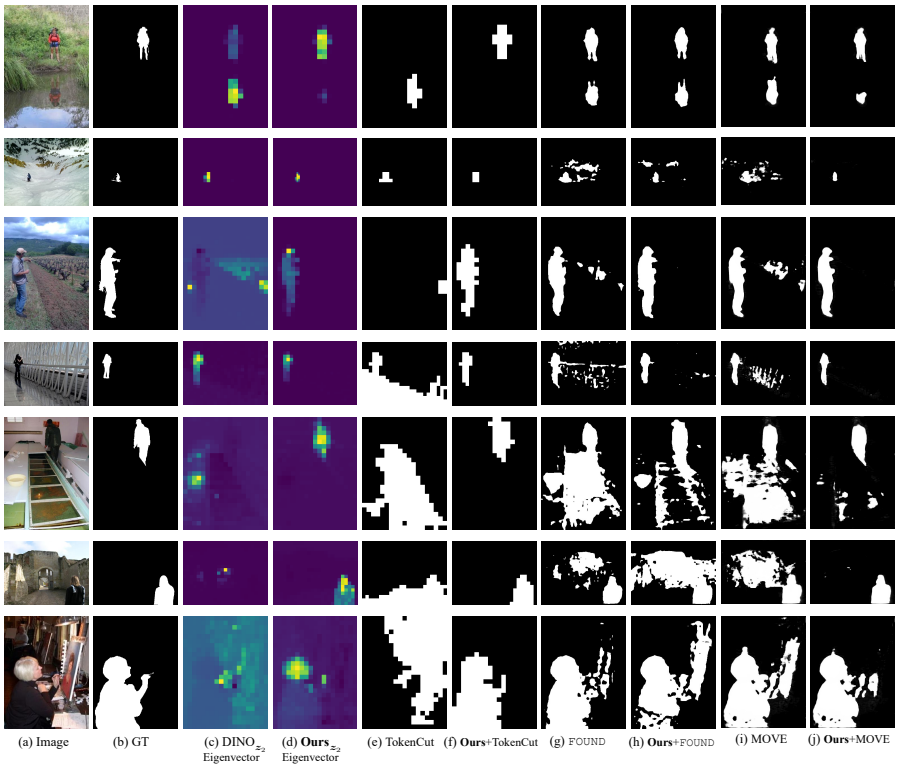

Fig. 3: More Visualization Results. Our method has a better and cleaner eigen attention mask (d), thus improving the performance of existing state-of-the-art methods by removing noisy backgrounds.

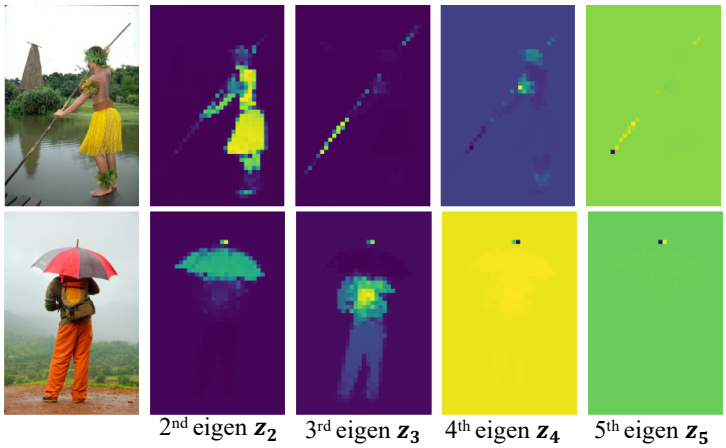

Fig. 4: Eigenvectors VS. Semantic segments