Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization

Supplementary Material

1. Implementation Details

Here, we present full implementations details for reproducing our results.

Spectral Decomposition We operate on images at their full resolution. For the spectral decomposition step, we start by extracting the key features $f \in \mathbb{R}^{C \times M/P \times N/P}$ from the final layer of the vision transformer. We then normalize these features along the embedding dimension and compute the affinity matrix W_{feat}. Next, we calculate color features W_{knn} by first downsampling the image to the intermediate resolution resolution $M' \times N'$ and then calculating the sparse KNN affinity matrix using the implementation from PyMatting [15]. We use $M' = M/8$ and $N' = N/8$ for all experiments.

Next we perform the fusion of color and feature information. If $P = 8$, then $\frac{MN}{P^2} = M' \cdot N'$, so the feature affinities above are already at the intermediate resolution. If $P = 16$, the feature affinities are upscaled 2× to size $M' \times N'$. The affinities are summed, weighted by λ_{knn} and the eigenvectors of the Laplacian L are calculated using the Lanczos algorithm.

We also need to address the fact that vision transformers only operates on images whose size is a multiple of the patch size P. Since we operate on images at their full resolution, which is not always a multiple of P, this presents a small complication. To resolve this, we simply crop images to the nearest multiple of P by truncating the right and top edges. For object localization, this means that we do not predict bounding box coordinates in the cropped edge regions. For the segmentation task, we compute segmentations on the cropped image and then simply replicate the right and bottom edges until the segmentation matches the original image. Fortunately, since the patch size is quite small relative to the image resolution, this cropping is not a large issue for the vast majority of images.

Object Localization For the object segmentation task, we consider the smallest eigenvector $y_1 \in \mathbb{R}^{\frac{MN}{P^2}}$ of L with nonzero eigenvalue. We reshape y_1 to size $\frac{M}{P} \times \frac{N}{P}$ and compute its largest fully-connected component. We then draw a bounding box (x_1, y_1, x_2, y_2) around this component and multiply its coordinates by P to obtain the bounding box in the scale of the original image.

Unsupervised single-object segmentation Single-object segmentation is a natural extension of our localization pipeline. As above, we first using the Fiedler eigenvector to find a coarse object segmentation. We then apply a pairwise CRF to increase the resolution of our segmentations back to the original image resolution (M, N). For the CRF, we use the implementation from [24] and leave all parameters on their default settings.

Semantic Segmentation For the semantic segmentation task, we consider the n smallest eigenvectors $\{y_i : i < K\}$ of L, reshaped into a tensor of size $n \times \frac{M}{P} \times \frac{N}{P}$. All experiments use $n = 15$. We cluster these eigenvectors using k-means clustering with $k = 15$ to obtain a discrete (non-semantic) segmentation of size $\frac{M}{P} \times \frac{N}{P}$ for each image; this breaks the image into k separate segments/regions. In each image, the largest segment is considered to be the background region. We then compute a feature vector for each of the $(k-1) \times T$ non-background segments in our dataset of size T. This feature vector is computed by taking a bounding box around each segment, expanding this bounding box by 2 patches, cropping this region of the image, and applying the self-supervised transformer to this cropped image. These segment features are clustered over the dataset via K-means clustering with $K = 20$ (which is the number of non-background classes in PASCAL VOC). Finally, associating each segment in each image with its cluster produces (low-resolution) semantic segmentations. We carry out the above steps on the `train_aug` set of PASCAL VOC.

For the self-training stage, we begin by upscaling each (low-resolution) semantic segmentation obtained in the previous step to the original image resolution. We train a ResNet-50 with a pretrained DINO [6] backbone and a DeepLab [7] head. We train for 2000 steps with the Adam [22] optimizer, learning rate $1 \cdot 10^{-4}$, batch size 144, and random crops of size 224. We decay the learning rate to zero linearly over the course of training. This setup was not extensively tuned and could likely be improved with a comprehensive hyperparameter sweep. Finally, we evaluate this model on the `val` set of PASCAL VOC.

Computational Requirements All experiments are performed on a single NVidia GPU with 16GB of memory. The eigenvector computation is performed on the CPU and takes approximately 0.5s for an image of size 512px at intermediate resolution $H' = W' = H/8 = W/8 = 32$px. Training the semantic segmentation network from the pseudolabeled images takes approximately 2 hours to finish 2000 steps. These low computational requirements are one of the strengths of our method, and make it a sensible baseline for future work.

2. Additional Qualitative Examples

We show additional qualitative examples of our method, including the extracted eigenvectors (Fig. 7) and results on single-object localization (Fig. 8) and single-object segmentation (Fig. 9). Finally, for the semantic segmentation task, we show the classes discovered by our approach. It is important to note that, since our approach is fully unsupervised and uses self-supervised features, eigensegment clusters do not necessarily align with the semantic categories annotated in the dataset (in this case, PASCAL VOC). In Fig. 10 we show the $K = 21$ categories found in the DINO feature space by our method and observe that, while some align with the ground truth label set (e.g., bus, dog, airplane, train), others do not (e.g., cluster 2 seems to be 'animal heads' instead of a species-specific cluster).

3. Additional Experiments and Ablations

In this section, we perform additional experiments and ablations in order to better understand the strengths and weaknesses of our method.

Ablation: Color Information To understand the importance of adding color information to the features we evaluate the performance for different values of λ_{knn} in Tab. 7. When $\lambda_{\text{knn}} = 0$, the normalized Laplacian matrix L is composed purely of semantic information. When $\lambda_{\text{knn}} \to \infty$, the normalized Laplacian matrix L is composed purely of color information. Larger models and variants with smaller patches benefit less from additional color information as their features likely inherently contain more local image information. Using color features alone is ineffective.

Ablation: Feature Type In Tab. 6, we present results when features are extracted from different parts of a self-attention layer. Features from attention *keys* perform best by a large margin, in line with the intuition that the key projection layer should align keys into a shared space for subsequent comparison with query vectors.

Ablation: Feature Depth In Tab. 8, we present results when features are extracted from different blocks of a ViT network. We find that features from later blocks perform better; they contain semantic information which is both spatially-localized and easy to extract with spectral methods.

Ablation: Network Architecture In Tab. 9, we present results when features are extracted from three new architectures: ResNet-50, ConvNext [26], and XCit [12]. ConvNext is a purely-convolutional network pretrained (in a supervised manner) on ImageNet, and it may be thought of as an improved ResNet. XCiT is a transformer with cross-covariance attention (i.e. attention over features rather than spatial locations) pretrained using DINO. ConvNext substantially outperforms ResNet-50 under this setup, demonstrating a link between classification performance and unsupervised object localization performance. Nonetheless, both ConvNext and XCiT lag behind the strong performance of the standard ViT, suggesting that the well-localized nature of *spatial self-attention* is one of the keys to the success of our deep spectral segmentation approach.

Ablation: Semantic Segmentation Pipeline In Tab. 10 and Tab. 11, we present ablation results for varying two aspects of our semantic segmentation pipeline: the number of eigenvectors used in the first stage, and the number of clusters K used in the second stage. We see that our method is fairly robust to the number of eigenvectors used in the first stage, unless one uses a very small number of eigenvectors (i.e. fewer than three). For the number of clusters K, we first note that the value used in the main paper is not chosen empirically, but rather set to the number of classes (incl. background) in the PASCAL VOC dataset, as is common for evaluation purposes. In the ablation, for $K > 20$ (i.e. over-clustering), we compute the optimal matching between our predictions and the ground-truth classes. Thus, larger K yields superior mIoU scores.

Additional Experiment: Class-Agnostic Detection In Tab. 12, we give results for a slightly modified detection setting previously explored in [33]. In this setting, denoted class-agnostic detection, we train a class-agnostic object detection model by using the bounding boxes obtained from our method as pseudo-labels. For fair comparison, we follow the same training and evaluation procedure as LOST, including all hyperparameters. We see that our method outperforms LOST despite not tuning any hyperparameters for this task.

4. Discussion of Failure Cases

Here, we show examples of failure cases and discuss their potential causes, with the goal of fascilitating future research into unsupervised segmentation.

Feature	CorLoc
Final attention key (k)	**61.6**
Final attention query (q)	33.1
Final attention value (v)	49.9
Final attention output (o)	37.3

Table 6. **Feature type ablation.** Single-object localization performance using different features of a DINO-pretrained ViT-S model, evaluated on PASCAL VOC 2007. Key features perform slightly better than value and much better than query or output features.

λ_{knn}	**0.0**	**1.0**	**8.0**	**10.0**	**inf.**
ViT-S-16	58.0	60.1	**61.9**	61.5	-
ViT-B-16	57.7	59.5	61.1	**61.2**	24.2
ViT-S-8	59.4	60.0	**62.6**	62.5	-
ViT-B-8	60.5	61.2	**62.7**	62.5	-

Table 7. **Importance of color information.** An ablation of single-object model localization performance for different values of λ_{knn} on PASCAL VOC 2012. We find that larger models benefit less from color information, similar to models with smaller patches, as their features likely contain more color information per se.

Block	mIoU
12	**61.6**
11	61.4
8	50.5
4	28.2

Table 8. **Ablation across ViT blocks.** Single-object localization performance (CorLoc) on PASCAL VOC 2007 using features extracted from different blocks of a ViT-s16 (DINO) model. Note that the model has 12 blocks, so 12 refers to the last block.

Arch.	Feature	CorLoc
ConvNext	Last conv. in stage 2	**41.8**
ConvNext	Last conv. in stage 3	40.7
ConvNext	Last block in stage 2	38.8
ConvNext	Last block in stage 3	31.2
XCiT	Cross-covariance attention key	33.6
ResNet-50	Last block	26.6

Table 9. **Ablation across new architectures.** Single-object localization performance (CorLoc) on PASCAL VOC 2007 using features extracted from ConvNext [26] and XCit [12].

Num. Eigs	mIoU	(w/ ST) mIoU
3	29.7	32.7
5	**33.3**	36.3
10	31.4	**37.5**
15	31.8	36.0

Table 10. **Ablation: Number of eigenvectors for semantic segmentation.** We vary the number of eigenvectors m used in the first step of our semantic segmentation pipeline. Using only three eigenvectors performs poorly, as they are not always sufficient to differentiate different segments of complex scenes. Above three eigenvectors, our method is not very sensitive to the exact number of eigenvectors used.

K	mIoU
20	33.3
30	38.5
40	42.8

Table 11. **Ablation: Value of K for semantic segmentation.** We vary the number of clusters K used in the second step of our semantic segmentation pipeline. For $K > 20$ (i.e. over-clustering), we compute the optimal matching between our semantic clusters and the ground-truth classes. Note that, as a result, larger K yield superior mIoU numbers. We report results without self-training.

Method	CorLoc
LOST (w/ self-training)	64.5
Ours (w/ self-training)	**65.1**

Table 12. **Additional experiment: class-agnostic detection on VOC2007.** In this experiment, we train a class-agnostic object detection model by using the bounding boxes obtained from our method as pseudo-labels. For fair comparison, we follow the same training and evaluation procedure as LOST, which refers to this setup as CAD (class-agnostic detection).

Spectral Decomposition Although the notion of a failure case for eigenvectors is not well-defined, we will characterize a failure case as one in which the vectors produced by our method do not match up with our human intuition about the major objects in the scene. We show examples in Fig. 11. These failure cases often occur when a very small object in the foreground lies in the plane of the image, for example in the last row of the figure. In these cases, the first eigensegment will usually segment this small region. Another failure case in PASCAL VOC occurs when images have borders or frames (these images are present due to the web-scraped nature of the dataset). In these cases, the model nearly always identifies the frame in its first eigenvector.

Object Localization We show examples of failure cases for the object localization task in Fig. 11. When our spectral segmentation method fails, it is usually the result of locating a group of semantically related objects (*e.g.* a group of people) rather than a single entity (*e.g.* an individual person). We note, however, that in many cases these instances are indeed separated by the latter eigenvalues (see Fig. 3 in the main paper); utilizing this information to separate object instances could be an interesting avenue for future research.

Semantic Segmentation We show examples of failure cases for the semantic segmentation task in Fig. 11. We see that the network sometimes fails to detect multiple distinct semantic regions in the same image. Qualitatively, we have observed that this failure mode is actually more common after self-training. In other words, self-training seems to improve performance overall by improving the quality of individual masks, but also seems to hurts the models' ability to segment multiple regions in the same image. There are also some failure cases in which our network should have sharper object boundaries, as is the case with most segmentation networks.

5. Additional Related Work

Here, we discuss relevant work that could not be included in the main paper due to space constraints.

Unsupervised Segmentation Current methods for unsupervised semantic segmentation can broadly be characterized as either generative or discriminative approaches.

For single-object segmentation, generative methods currently rank as the most active research direction, with numerous works having been proposed in the last two years [3, 4, 8, 19, 20, 29, 39, 40]. Most commonly, these methods work by generating images in a layerwise fashion and compositing the results. For example, ReDO [8] uses a GAN to re-draw new objects on top of existing objects/regions, and Copy-Paste GAN [2] copies part of one image onto the other. Labels4Free [1] trains a StyleGAN to generate images in a layer-wise fashion, from which a segmentation may easily be extracted. [29,39] extract segmentations from pretrained generative models such as BigBiGAN. However, these generative approaches are severely limited in that they only perform foreground-background segmentation: they can only segment a *single* object in each image, and most involve training new GANs. As a result, they are not well-suited to segmenting complex scenes nor to assigning semantic labels to objects. Another family of methods, most of which adopt a variational approach [23], focus on unsupervised scene decomposition, effectively segmenting multiple objects in an image [5, 11, 13, 14, 16, 25, 27, 30]. However, these methods cannot assign semantic categories to objects and struggle significantly on complex real-world data [16, 21].

Discriminative approaches are primarily based on clustering and contrastive learning. Invariant Information Clustering (IIC) [18] predicts pixel-wise class assignments and maximizes the mutual information between different views of the same image. SegSort [17] maximizes within-segment similarity and minimizes cross-segment similarity by sorting and clustering pixel embeddings. Hierarchical Grouping [41] performs contour detection, recursively merges segments, and then performs contrastive learning. MaskContrast [38], the current state of the art in unsupervised semantic segmentation, uses saliency detection to find object segments (*i.e*. the foreground) and then learns pixel-wise embeddings via a contrastive objective. However, MaskContrast relies heavily on a saliency network which is initialized with a pretrained (fully-supervised) network. However, it relies on the assumption that all foreground pixels belong to the same object category, which is not necessarily the case. As a result, it is limited to predicting a single class per image without further finetuning. Finally, DFF [10] uses pre-trained features and performs non-negative matrix factorization for co-segmentation.

6. Further Description of the Laplacian

In this section, we present a slightly extended description of the spectral graph theoretic methods that underly our paper.

Consider a connected, undirected graph $G = (V, E)$ with edges E and vertices V. We denote by $W = (w_{ij})$ the edge weights between vertices i and j. In our case, G corresponds to an image I, where the vertices V are image patches and the edge weights W are defined by the semantic affinities of patches.

Let $f : V \to \mathbb{R}$ be a real-valued function defined on the vertices of G. Note that these functions are synonymous with vectors of length V, and are thus also synonymous with segmentation maps. We begin by asking the question: what does it mean for a function f to be *smooth* with respect to the graph G?

Intuitively, f is smooth when its value at a vertex is similar to its value at each of the vertex's neighbors. If we quantify this similarity using the sum of squared errors, we obtain:

$$\sum_{(i,j)\in E} (f(i) - f(j))^2 \tag{1}$$

which is a symmetric quadratic form. This means that there exists a symmetric matrix L such that

$$x^T L x = \sum_{(i,j)\in E} (x_i - x_j)^2$$

for $x \in \mathbb{R}^n$ with $n = |V|$. This matrix L is the called Laplacian of G.

While there are many ways to define L, we present this definition because we believe it gives the greatest insight into the success of our method. The standard way of defining L is by the formula $L = D - W$, where $D_{ii} = \deg(i)$ is the diagonal matrix of row-wise sums of W. The quadratic form definition makes clear some of the fundamental properties of the Laplacian: L is symmetric and positive semi-definite, since $x^T L x \geq 0$ for any x. Additionally, its the smallest eigenvalue is 0, corresponding to a (non-zero) constant eigenfunction.

To gain intuition for the Laplacian, we present a very simple example from the domain of physics, adapted from [28]. Consider modeling a fluid which flows between a set of reservoirs (vertices) through pipes (edges) with different capacities. Physically, the fluid flow through an edge is proportional to the difference in pressure between its vertices, $x_i - x_j$. Since the total flow into each vertex equals the total flow out, the sum of the flows along a vertex i is 0:

$$\begin{aligned} 0 &= \sum_{j \in N(i)} x_j - \sum_{j \in N(i)} x_i = \deg(i) x_i - \sum_{j \in N(i)} x_j \\ &= ((D - W)x)_i = (Lx)_i \end{aligned}$$

This is known as the Laplace equation $Lx = 0$, and it is the simplest special case of Poisson's equation $Lx = h$.

The Laplacian spectrum is the centerpiece of our method. As described in the main paper, the eigenfunctions of L are orthonormal and form a basis for the space of bounded functions on G. The Laplacian spectrum does not fully determine the underlying graph, but it nonetheless contains a plethora of information about its structure. Our paper leverages this information for a variety of unsupervised dense computer vision tasks.

For further reading in spectral graph theory, we encourage the reader to look into the following resources: [9, 28, 34, 35].

7. Broader Impact

It is important to discuss the broader impact of our work with respect to methodological and ethical considerations.

From an ethical perspective, models trained on large-scale datasets — even the ones considered in our work which are trained without supervision — might reflect biases and stereotypes introduced during the image collection process [31, 32, 36, 37]. In addition, datasets such as ImageNet (used to train the models) and PASCAL VOC (used in our evaluations) contain images from the web (e.g., Flickr). This data is collected without consent and might also contain inappropriate content [31], which raises ethical and legal issues. Our method discovers concepts existing in the data through the lens of self-supervised pre-training and, as such, it may be implicitly affected by underlying biases. For this reason, our method should only be used for research purposes and not in any critical or production applications.

From a methodological perspective, our approach reflects the degree to which different object categories are encoded in the feature space of self-supervised learners. Since we do not fine-tune models on a specific dataset for a specific task (e.g., semantic segmentation), these predicted categories may differ from the pre-defined set of categories which are (somewhat arbitrarily) annotated in a given benchmark. For example, the categories found by the decomposition and clustering of DINO features, might not necessarily align with the ones annotated in PASCAL VOC; in fact, there is little reason why that should be besides some commonly occurring objects. Therefore, how to properly and fairly evaluate fully unsupervised algorithms remains an open question.

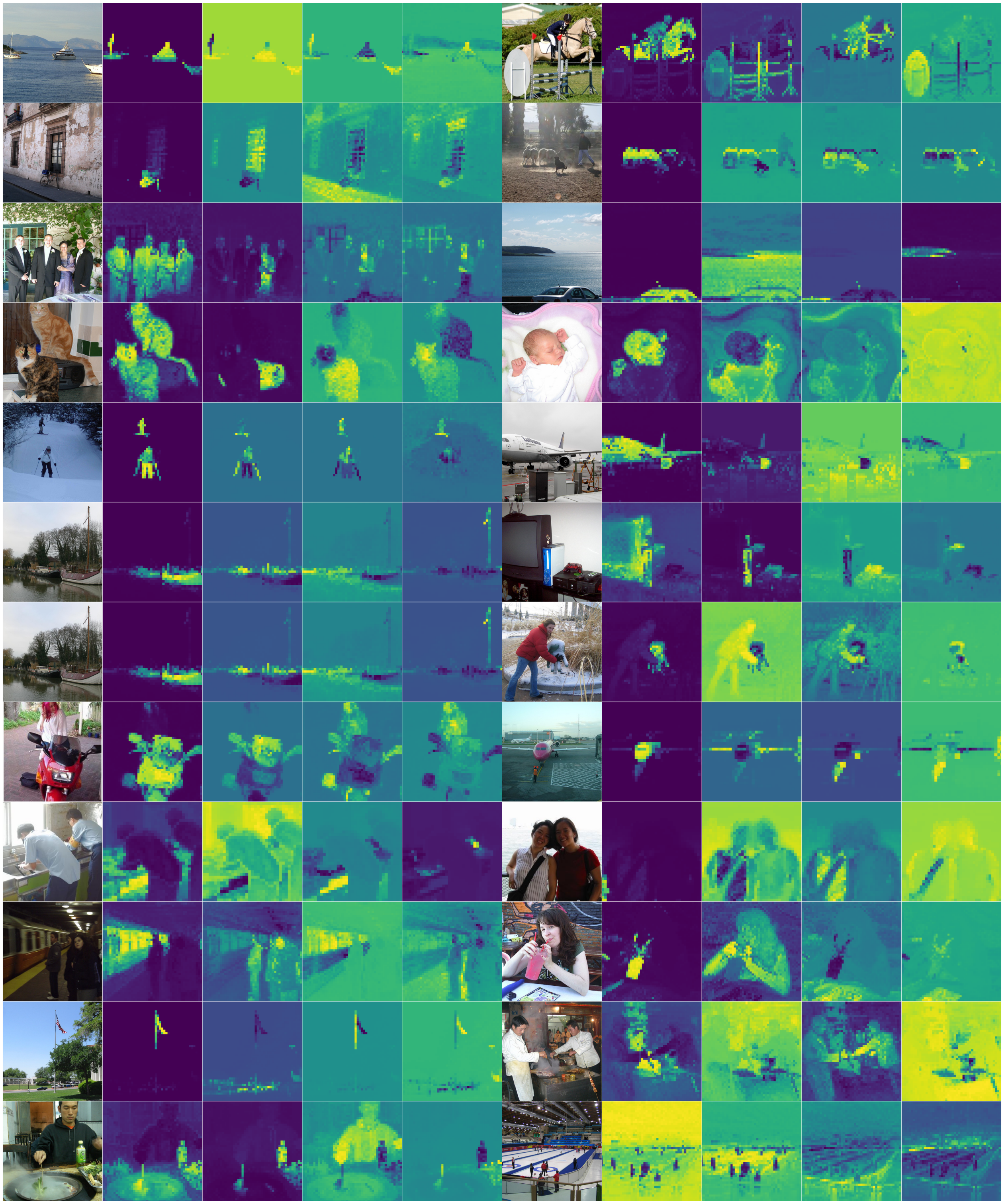

Figure 7. **Additional examples of eigenvectors extracted by our method on random images from PASCAL VOC 2012.** The first column in each column shows the original image, while the following three columns show the first three eigenvectors.

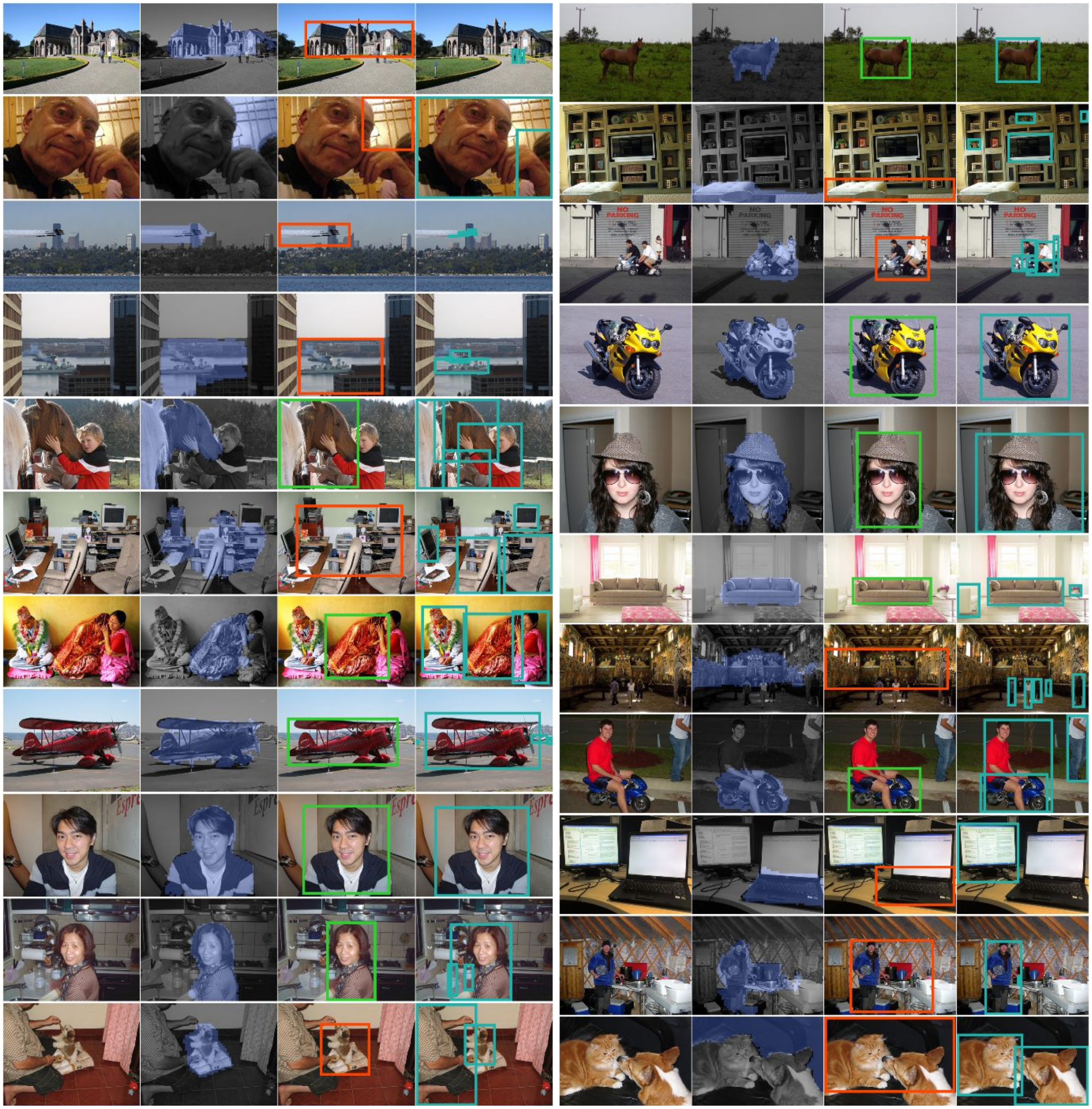

Figure 8. **Additional object localization examples of our method on random images from PASCAL VOC 2012.** The first column shows the original image, while the following three columns show our first eigenvector (thresholded at zero), our predicted bounding box, and the ground truth bounding box, respectively. Our bounding box is colored in green or red based on whether it has greater than 50% mIoU with one of the ground-truth bounding boxes.

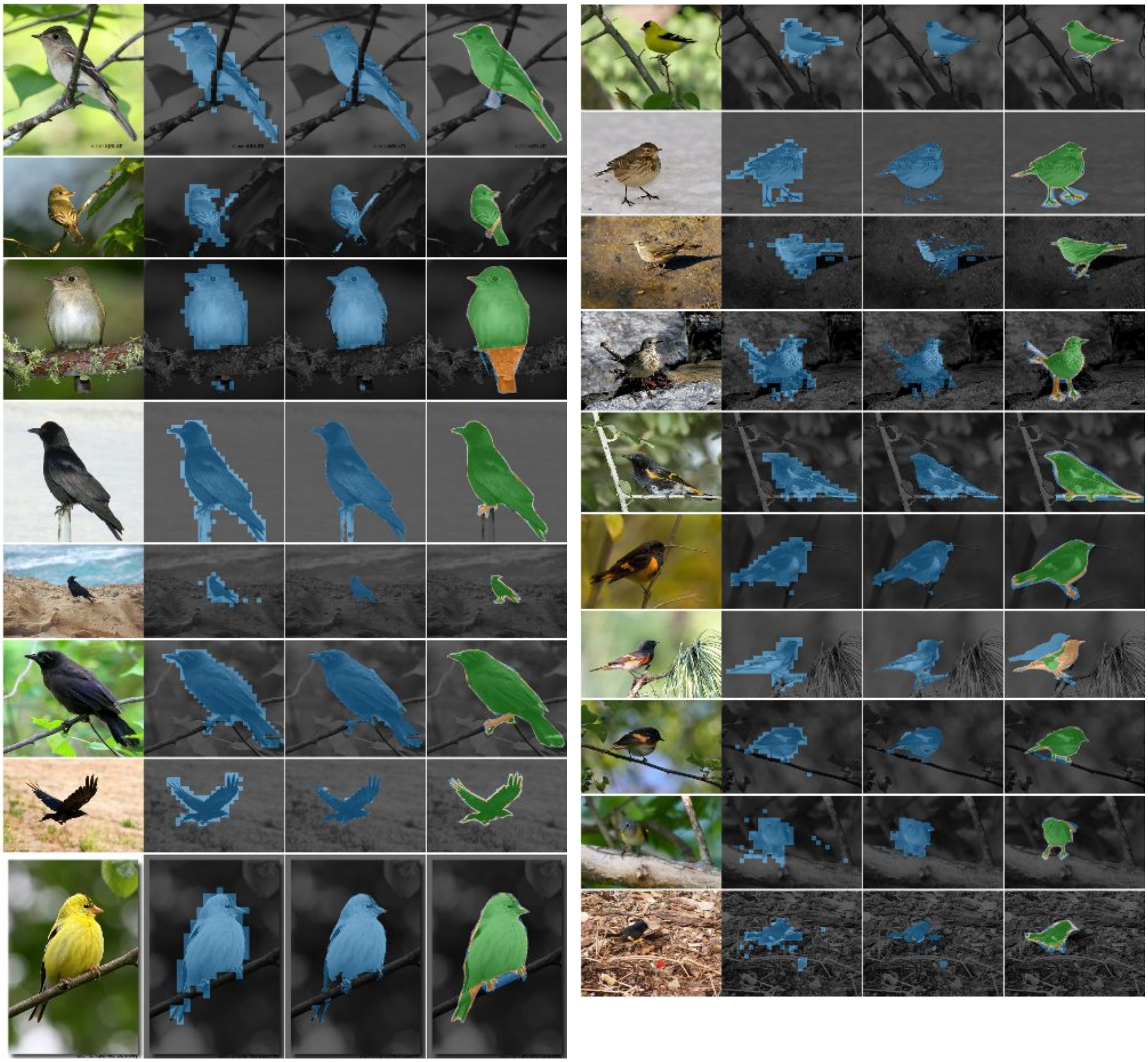

Figure 9. **Examples of our method for the single-object segmentation task on random images from CUB.** The first row in each column shows the original image, while the following three rows show our first eigenvector (thresholded at zero), our predicted segmentation, and the ground-truth segmentation. Our segmentation masks accurately locate the bird, often segmenting it without including other objects such as branches or leaves (which is a common failure point of prior state-of-the-art methods). Note that these images are not cherry picked in any way; they are the first images in the CUB dataset.

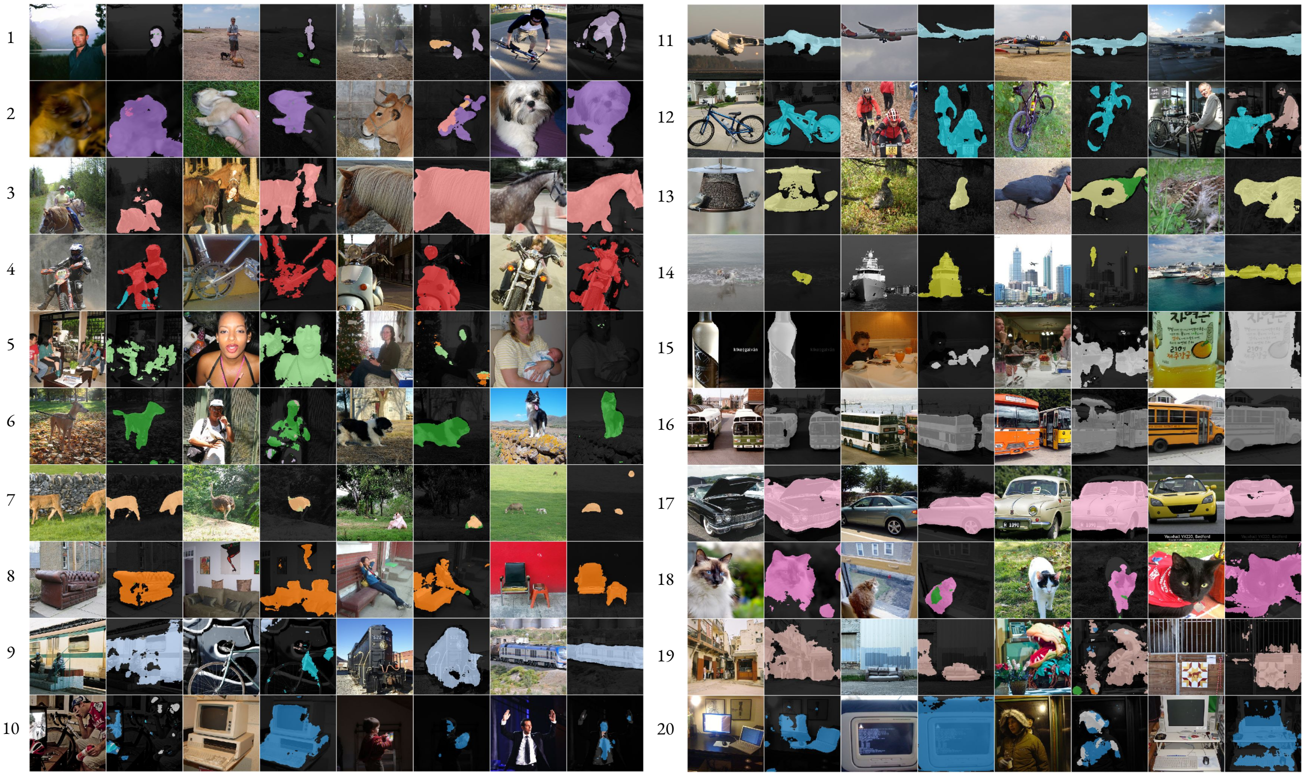

Figure 10. **Per-pseudoclass examples of our method for the semantic segmentation task on random images from the validation set of PASCAL VOC 2012.** For each pseudoclass (*i.e.* cluster), we show four randomly selected images from PASCAL VOC for which the given class is the largest segmented region in the object. We see that our pseudoclasses correspond to numerous identifiable concept such as people, buses, boats, cats, airplanes, and bicycles without any human supervision.

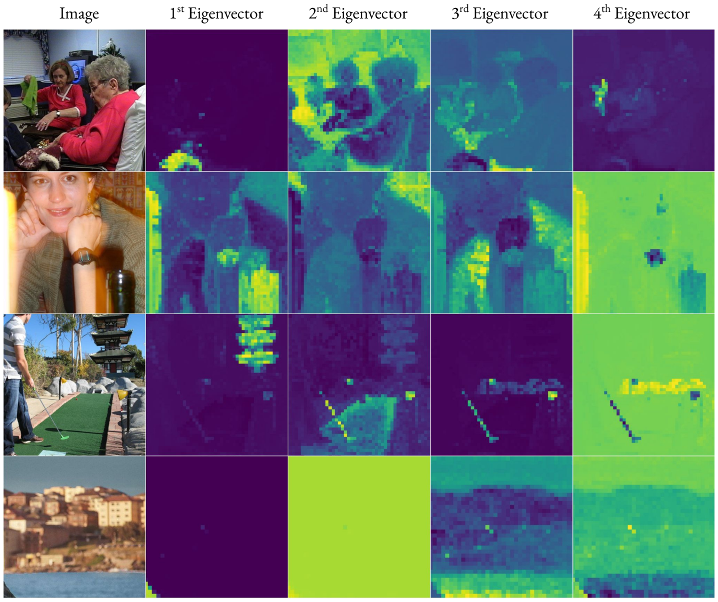

Figure 11. **Examples of failure cases for the eigensegments.** The eigenvectors of the feature Laplacian do not correspond to the primary objects and regions in the scene. These failure cases often occur when a very small object in the foreground lies in the plane of the image, for example in the last image above.

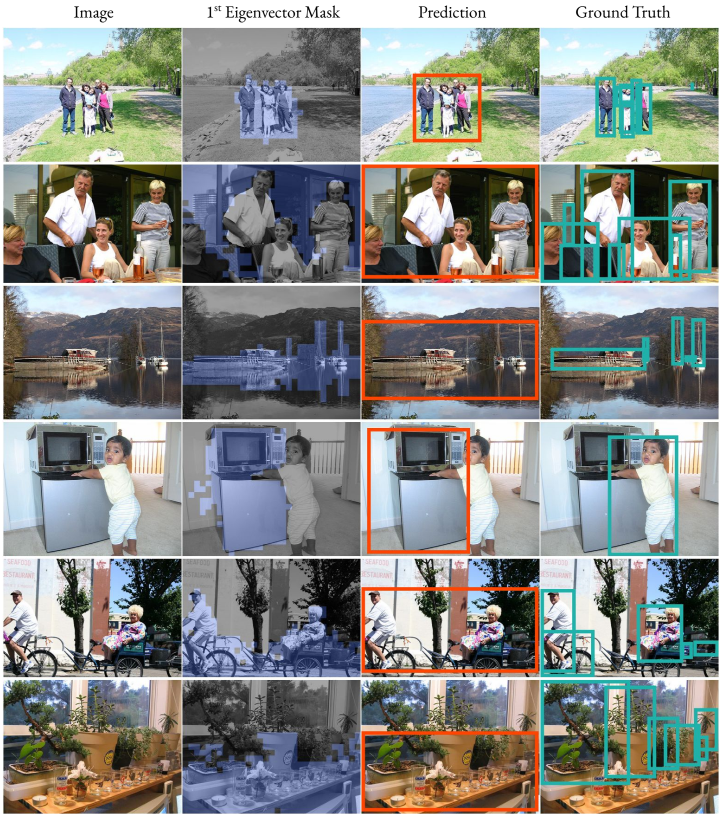

Figure 12. **Examples of failure cases for the object localization task.** When our spectral segmentation method fails, it is usually the result of locating a group of semantically related objects (*e.g*. a group of people) rather than a single entity (*e.g*. an individual person). We note, however, that in many cases these instances are indeed separated by the latter eigenvalues (see Fig. 1); utilizing this information to separate object instances could be an interesting avenue for future research.

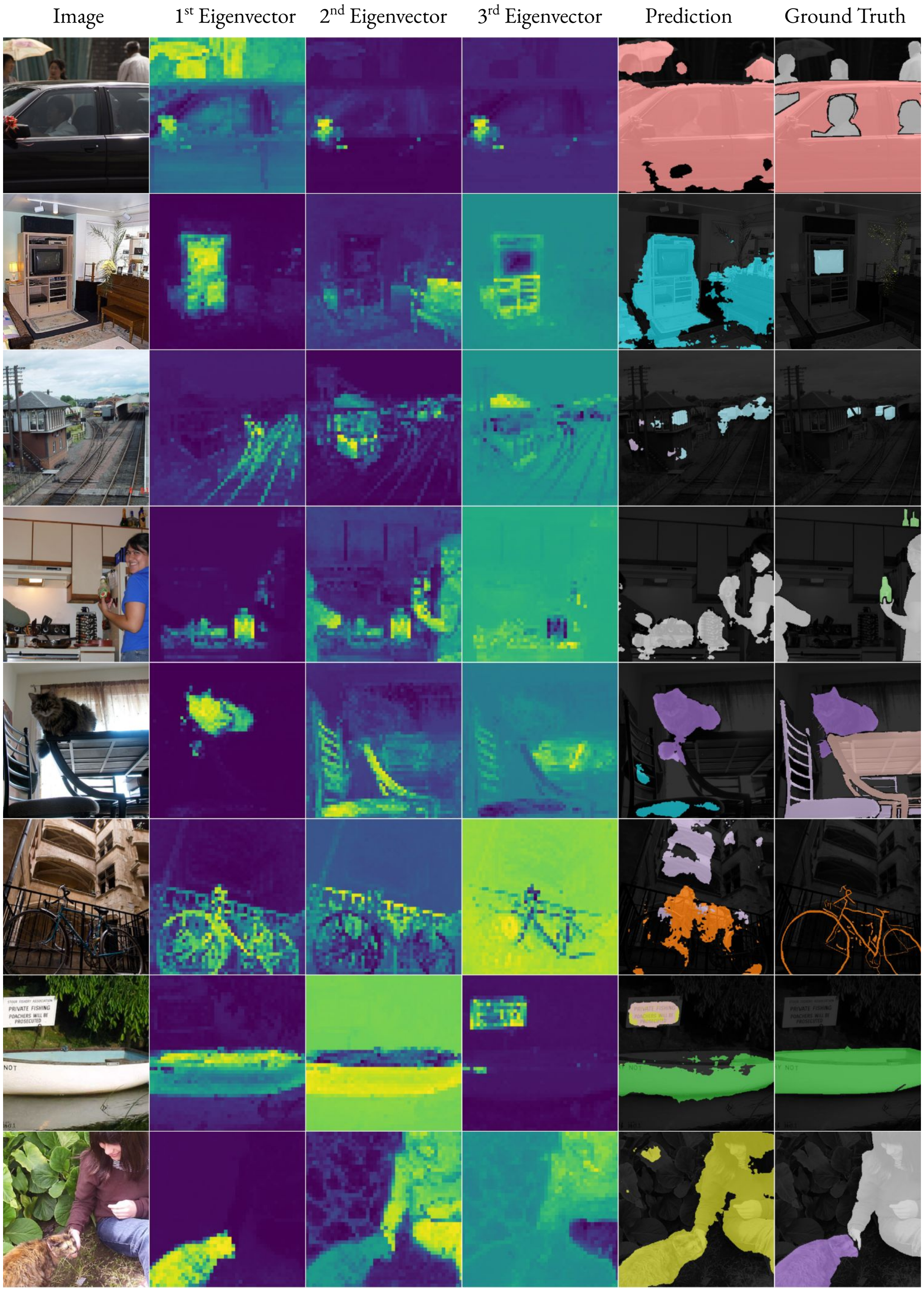

Figure 13. **Examples of failure cases for the semantic segmentation task.** The network sometimes fails to detect multiple distinct semantic regions in the same image. Qualitatively, we have observed that this failure mode is actually more common after self-training. In other words, self-training seems to improve performance overall by improving the quality of individual masks, but also seems to hurts the models' ability to segment multiple regions in the same image. There are also some failure cases in which our network should have sharper object boundaries, as is the case with most segmentation networks.

References

[1] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Labels4free: Unsupervised segmentation using stylegan. In *Proc. ICCV*, 2021. 4

[2] Relja Arandjelović and Andrew Zisserman. Object discovery with a copy-pasting gan. *arXiv preprint arXiv:1905.11369*, 2019. 4

[3] Yaniv Benny and Lior Wolf. Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. *Lecture Notes in Computer Science*, page 514–530, 2020. 4

[4] Adam Bielski and Paolo Favaro. Emergence of Object Segmentation in Perturbed Generative Models. In *Proc. NeurIPS*, volume 32, 2019. 4

[5] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 4

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[8] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised Object Segmentation by Redrawing. In *Proc. NeurIPS*, volume 32, 2019. 4

[9] Fan Chung. Lectures on spectral graph theory. *CBMS Lectures, Fresno*, 6:17–21, 1996. 5

[10] Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 336–352, 2018. 4

[11] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3412–3420. AAAI Press, 2019. 4

[12] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Herve Jegou. XCit: Cross-covariance image transformers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Proc. NeurIPS*, 2021. 2, 3

[13] Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2970–2981. PMLR, 18–24 Jul 2021. 4

[14] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*. OpenReview.net, 2020. 4

[15] Thomas Germer, Tobias Uelwer, Stefan Conrad, and Stefan Harmeling. Pymatting: A python library for alpha matting. *Journal of Open Source Software*, 5(54):2481, 2020. 1

[16] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2424–2433. PMLR, 2019. 4

[17] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7334–7344, 2019. 4

[18] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653*, 2018. 4

[19] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 4

[20] A. Kanezaki. Unsupervised Image Segmentation by Backpropagation. In *Proc. ICASSP*, pages 1543–1547, 2018. 4

[21] Laurynas Karazija, Iro Laina, and Christian Rupprecht. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 4

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[23] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 4

[24] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proc. NeurIPS*, 2011. 1

[25] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*. OpenReview.net, 2020. 4

[26] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proc. CVPR*, 2022. 2, 3

[27] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pages 11525–11538. Curran Associates, Inc., Dec. 2020. 4

[28] Luke Melas-Kyriazi. The mathematical foundations of manifold learning. *CoRR*, abs/2011.01307, 2020. 5

[29] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Finding an unsupervised image segmenter in each of your deep generative models. *arXiv preprint arXiv:2105.08127*, 2021. 4

[30] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 4

[31] Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv preprint arXiv:2006.16923*, 2020. 5

[32] Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D Sculley. No classification without representation: Assessing geodiversity issues in open data sets for the developing world. *arXiv preprint arXiv:1711.08536*, 2017. 5

[33] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *Proc. BMVC*, November 2021. 2

[34] Daniel Spielman. Spectral graph theory and its applications. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 29–38. IEEE, 2007. 5

[35] Daniel Spielman. Spectral graph theory. In *Combinatorial scientific computing*, number 18. Citeseer, 2012. 5

[36] Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 701–713, 2021. 5

[37] Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–512, 2018. 5

[38] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *International Conference on Computer Vision*, 2021. 4

[39] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Big gans are watching you: Towards unsupervised object segmentation with off-the-shelf generative models. *arXiv.cs*, abs/2006.04988, 2020. 4

[40] Xide Xia and Brian Kulis. W-net: A deep model for fully unsupervised image segmentation. In *arXiv.cs*, 2017. 4

[41] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. In *Proc. NeurIPS*, volume 33, 2020. 4