Learning Neural Eigenfunctions for Unsupervised Semantic Segmentation

Zhijie Deng
Shanghai Jiao Tong University
Shanghai, China
zhijied@sjtu.edu.cn

Yucen Luo
Max Planck Institute for Intelligent Systems
Tübingen, Germany
luoyucencen@gmail.com

Abstract

Unsupervised semantic segmentation is a long-standing challenge in computer vision with great significance. Spectral clustering is a theoretically grounded solution to it where the spectral embeddings for pixels are computed to construct distinct clusters. Despite recent progress in enhancing spectral clustering with powerful pre-trained models, current approaches still suffer from inefficiencies in spectral decomposition and inflexibility in applying them to the test data. This work addresses these issues by casting spectral clustering as a parametric approach that employs neural network-based eigenfunctions to produce spectral embeddings. The outputs of the neural eigenfunctions are further restricted to discrete vectors that indicate clustering assignments directly. As a result, an end-to-end NN-based paradigm of spectral clustering emerges. In practice, the neural eigenfunctions are lightweight and take the features from pre-trained models as inputs, improving training efficiency and unleashing the potential of pre-trained models for dense prediction. We conduct extensive empirical studies to validate the effectiveness of our approach and observe significant performance gains over competitive baselines on Pascal Context, Cityscapes, and ADE20K benchmarks. The code is available at `https://github.com/thudzj/NeuralEigenfunctionSegmentor`.

1. Introduction

Semantic segmentation is essential in understanding the inherent structure and fine-grained information of images. However, current approaches often hinge on a vast amount of manual annotations to train neural networks (NNs) effectively in an end-to-end manner [5, 40]. This is problematic, as obtaining these annotations can be both time-consuming and costly, particularly in fields such as autonomous driving and medical image processing, where annotations are usually collected from domain experts. Thus, finding a way to perform semantic segmentation without manual annotation remains a crucial unresolved problem.

Unsupervised semantic segmentation has recently attracted a great deal of attention. A number of methods attempt to tackle it by learning fine-grained image features using self-supervised objectives and then applying clustering or grouping techniques [46, 41]. They tend to recognize single objects or single semantic categories and struggle with complex images. Other approaches have tried to use vision-language cross-modal models (e.g., CLIP [35]) to achieve zero-shot semantic segmentation [48, 39], but they heavily rely on carefully-tuned text prompts and self-training. Compared to the recent approaches, the classic spectral clustering [38], which has stood the test of time, remains an appealing option. In particular, it enjoys solid foundations in spectral graph theory—it finds the minimum cut of the connectivity graph over pixels.

However, traditional spectral clustering exhibits limitations in three aspects: (*i*) it operates on raw image pixels, thus is sensitive to color transformations and unable to recognize semantic similarities; (*ii*) it is computationally inefficient due to the involved spectral decomposition; (*iii*) unlike NN-based methods, it is nontrivial and costly to extend to non-training samples because of its transductive manner. Thus it cannot be performed end-to-end in the test phase. Recent work reveals that pre-trained models such as ViTs [14] can mitigate the first limitation, significantly improving the applicability and effectiveness of spectral clustering [30]. Its core contribution is to build the connectivity graph over image patches based on an affinity matrix computed with the dense features from pre-trained models. Still, the limitations regarding efficiency and flexibility remain.

The present paper aims to overcome the remaining limitations, rendering spectral clustering a simple yet effective baseline for unsupervised semantic segmentation. To tackle the inefficiency issue, we propose to cast the involved spectral decomposition problem as an NN-based optimization one using the recently developed neural eigenfunction (NeuralEF) technique [12]. Concretely, we first measure the similarities between image patches using both the features extracted from pre-trained models and raw pixels. Treating the similarity matrix (or its variants) as the output of a kernel function, we then optimize NNs to approximate its principal eigenfunctions. Consequently, our method constitutes an NN-based counterpart of spectral embedding. We eliminate the need for an additional grouping step, which is required in prior work [30], by constraining the NN output to one-hot vectors that indicate clustering assignments directly. To accomplish this, we use the Gumbel-Softmax estimator [20] for gradient-based optimization during training. These strategies transform spectral clustering from a non-parametric approach to a parametric one, enabling easy and reliable out-of-sample generalization and avoiding solving complex matrix eigenvalue problems during testing.

We perform extensive studies to evaluate the effectiveness of our approach for unsupervised semantic segmentation. We first experiment on the popularly used benchmarks Pascal Context [31] and Cityscapes [9] based on pre-trained ViTs, and report superior results compared to leading methods MaskCLIP [48] and ReCo [39]. We further consider the sliding-window-based evaluation protocol [40] and experiment on the challenging ADE20K dataset [47] to systematically study the behavior of our method. In addition, we conduct thorough ablation studies to gain insights into the specification of several core hyper-parameters.

2. Related Work

Unsupervised segmentation. Image segmentation has numerous practical applications in various industries and scientific fields. In order to alleviate the burden of collecting annotations, prior works have comprehensively studied the problem of learning image segmenters in semi- and weakly-supervised settings [24, 18, 4, 23]. Recently, there have been increasing efforts to tackle unsupervised segmentation based on the progress in related fields like deep generative models (DGMs) and self-supervised learning (SSL). On the one hand, DGM-based segmentation approaches train specialized image generators to separate foreground from background [6, 2, 1] or extract saliency masks directly from pre-trained generators [42]. Yet, it is technically nontrivial to extend them to cope with *semantic* segmentation. On the other hand, SSL-based methods define objectives to perform clustering [19, 7], mutual information maximization [21, 33], contrastive learning [41] or feature correspondence distillation [16] to learn image features suitable for grouping. However, most of these methods only recognize single objects or single semantic categories and struggle with complex images. With the increasing accessibility of pre-trained image-text cross-modal models, considerable efforts have been devoted to performing zero-shot semantic segmentation with them [48, 39]. Nevertheless, the entanglement with cross-modal models places high demands on the quality of the text prompts, which correspond to the semantic categories of concern, and hinders the methods from choosing backbone models freely.

Spectral clustering. As a classic solution to image segmentation, spectral clustering [38, 32] frames the original problem as a graph partitioning one defined on the connectivity graph over image pixels. Typically, spectral clustering exploits the eigenvectors of graph Laplacians to construct minimum-energy graph partitions [13, 15]. Spectral clustering is closely related to Kernel PCA [37] as they are both learning eigenfunctions [3]. Recently, spectral clustering has been combined with pre-trained models to enjoy rich semantic information [30]. Yet, it remains unsolved that spectral decomposition is expensive for big data, and the non-parametric nature hinders out-of-sample generalization. This paper aims to address these issues.

The deep learning variant of spectral methods. Refurbishing spectral methods with deep learning techniques is beneficial to improve the scalability and flexibility of the former. The spectral inference networks (SpIN) [34] is a seminal work in this direction. Yet, the learning objective of SpIN is ill-defined, which leads to only the subspace spanned by the principal eigenfunctions instead of the eigenfunctions themselves. SpIN hence introduces convoluted and expensive strategies to solve this problem. The recent NeuralEF technique [12] alternatively defines a new series of objective functions to break the symmetry among eigenfunctions explicitly. NeuralEF is further enhanced by weight sharing and extended to handle indefinite kernels in [11], constituting a more amenable choice for learning spectral embeddings given pre-defined kernels. We also note that there are several attempts to develop deep spectral clustering methods based on supervisions [28] or a dual autoencoder network [45], but they cannot be trivially applied to the task of unsupervised semantic segmentation due to the absence of annotations or other inefficiency issues.

3. Methodology

We begin with a brief review of the relevant background and then build up our approach step by step. We provide an overview of the proposed method in Figure 1.

3.1. Background

Semantic segmentation essentially targets determining the semantic consistency among image pixels. In the absence of annotations, the problem boils down to an unsupervised clustering one. Spectral clustering [38, 32] is a long-standing solution to it with solid theoretical foundations. It proceeds by partitioning a connectivity graph over image pixels. The resulting algorithm typically involves eigen-decomposing the graph Laplacian matrix and stacking its eigenvectors, which are then used in a Euclidean clustering algorithm to obtain a fixed number of partitions.

Traditional spectral clustering operates on raw image pixels, making it sensitive to color transformations and unable to identify semantic similarities. To address this, a

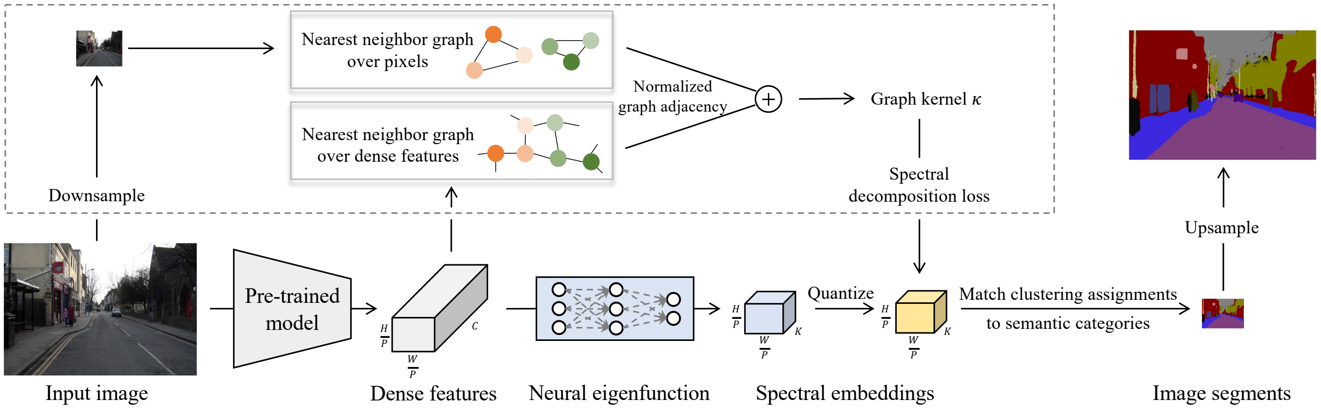
Nearest neighbor graph over pixels
Nearest neighbor graph over dense features
Normalized graph adjacency
Graph kernel κ
Spectral decomposition loss
Downsample
Upsample
Pre-trained model
H/P
W/P
C
K
Quantize
Match clustering assignments to semantic categories
Input image
Dense features
Neural eigenfunction
Spectral embeddings
Image segments

Figure 1. Method overview. We establish an end-to-end NN-based pipeline for spectral clustering to perform unsupervised semantic segmentation. The box highlights modules that exist only during training. The nearest neighbor graph over pixels only has edges between pixels from the same images, while the other one is built over dense features from various images. The pre-trained model is *fixed*.

natural idea is to include the inductive bias of NNs. In this spirit, the deep spectral method (DSM) [30] leverages powerful pre-trained models to translate the learning from raw pixels to patch-wise features that embed rich local semantics and are widely applicable. Specifically, let $\boldsymbol{x}_i \in \mathbb{R}^{H\times W\times 3}$ denote an image and $f : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H/P\times W/P\times C}$ denote the pre-trained model with P as a down-sampling factor. DSM constructs a semantic affinity matrix using both the patch-wise features $\boldsymbol{f}_{i,j,k} = f(\boldsymbol{x}_i)_{j,k} \in \mathbb{R}^C$ and image pixels, and performs spectral clustering on the corresponding connectivity graph. Subsequently, DSM performs bi-linear interpolation to convert patch-wise segments to pixel-wise ones. This simple workflow has surpassed various competitors regarding unsupervised semantic segmentation performance.

Denote by N the size of the dataset of concern. Semantic segmentation requires classifying pixels in different images yet with the same semantics into the same category. Thereby, it needs to ensure semantic consistency across the dataset. Therefore, DSM should, in principle, work on an affinity matrix of size $\mathbb{R}^{NHW/P^2\times NHW/P^2}$, but this is computationally infeasible as the involved spectral decomposition has a cubic complexity w.r.t. NHW/P^2. DSM alternatively conducts spectral clustering separately for each image and then performs cross-image synchronization, but this is inflexible compared to NN-based pipelines and arguably suboptimal. Additionally, existing spectral clustering methods, including DSM, are non-parametric, relying on costly spectral decomposition when faced with new test data.

In order to address the limitations discussed above, we leverage a parametric approach to spectral decomposition based on the recent NeuralEF technique [12] and learn discrete neural eigenfunctions directly. These innovations help to scale our method to large datasets, enable the straightforward and cheap out-of-sample extension for test data, and establish an NN-based spectral clustering workflow.

3.2. From Eigenvectors to Eigenfunctions

Spectral clustering involves the spectral decomposition of a matrix and hence is seemingly incompatible with the parametric NN models. To bridge this gap, we move our viewpoint from the eigenvectors to eigenfunctions.

Specifically, abstracting the graph Laplacian matrix as the evaluation of some kernel function κ on the dense features, the eigenfunctions of κ form a function-space generalization of the aforementioned eigenvectors. Formally, the eigenfunction ψ of a kernel $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ satisfies that

$$\int \kappa(\boldsymbol{x}, \boldsymbol{x}')\psi(\boldsymbol{x}')p(\boldsymbol{x}')d\boldsymbol{x}' = \mu\psi(\boldsymbol{x}), \tag{1}$$

where μ is the corresponding eigenvalue and p a probability measure. By definition, the eigenfunction should be normalized, and different eigenfunctions are orthogonal.

As shown, the eigenfunction takes input from the original space and maps it to the eigenspace specified by the kernel, where the local neighborhoods on data manifolds are preserved. Naturally, we can incorporate NNs as function approximators to the eigenfunctions. In this way, we bypass the need for expensive spectral decomposition of a matrix and can easily perform out-of-sample extension thanks to the generalization ability of NNs. Fortunately, the recently proposed NeuralEF [12] offers tools to realize this. We also note that the effectiveness of the spectral embedding yielded by neural eigenfunctions has been empirically validated in self-supervised learning [11].

3.3. Setting up the Graph Kernel

To set up the kernel for spectral clustering, as suggested by DSM [30], we construct two graphs with patch-wise features from pre-trained models and down-sampled image pixels, respectively. We then define the kernel with the weighted sum of the corresponding normalized graph adjacency matrices. By doing so, the high-level semantics and low-level details are conjoined.

Denote by $\mathbf{F} = \{\boldsymbol{f}_i\}_{i=1}^N \in \mathbb{R}^{NHW/P^2 \times C}$ the collection of patch-wise features over the dataset. Considering that the feature space shaped by large-scale pre-training is highly structured where simple distance measures suffice to represent similarities, we leverage cosine similarity to specify a nearest neighbor graph over image patches. The corresponding affinity matrix is detailed below:

$$\mathbf{A}_{u,v} = \begin{cases} \mathbf{F}_u \mathbf{F}_v^\top / (\|\mathbf{F}_u\| \|\mathbf{F}_v\|), & v \in k\text{-NN}(u, \mathbf{F}, \text{cosine}) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $k$-NN$(u, \mathbf{F}, \text{cosine})$ denotes the set of the $k$ nearest neighbors of $\mathbf{F}_u$ over $\mathbf{F}$ under cosine similarity. The above graph deploys edges between patches that are semantically close. The corresponding graph partitioning can result in meaningful segmentation at a coarse resolution. Although other traditional graphs can also be used, the nearest neighbor graph enjoys sparsity, which reduces storage and computation costs and has been studied extensively in manifold learning and spectral clustering.

To supplement the high-level features with low-level details, we bilinearly down-sample the original image $\boldsymbol{x}_i$ to $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^{H/P \times W/P \times 3}$ to keep resolution consistency, and fuse the spatial information (j, k) and color information $\tilde{\boldsymbol{x}}_{i,j,k}$ as a single vector. We stack the collection over the dataset as $\tilde{\mathbf{X}} \in \mathbb{R}^{NHW/P^2 \times 5}$. A nearest neighbor graph over down-sampled pixels is then defined based on L^2 distance following DSM [30]. The affinity matrix is:

$$\tilde{\mathbf{A}}_{u,v} = \begin{cases} 1, & v \in \tilde{k}\text{-NN}(u, \tilde{\mathbf{X}}, L^2) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\tilde{k}$-NN$(u, \tilde{\mathbf{X}}, L^2)$ denotes the set of $\tilde{k}$ nearest neighbors of $\tilde{\mathbf{X}}_u$ over $\tilde{\mathbf{X}}$ under L^2 distance (dissimilarity). We place a further constraint that the nearest neighbors should belong to the same image as the query to avoid establishing meaningless connections. It is expected that such a graph helps to detect sharp object boundaries.

We symmetrize $\mathbf{A}$ and $\tilde{\mathbf{A}}$ so that they can serve as graph adjacency matrices. Considering that the normalized graph cuts are more practically useful [38], we would better build a kernel with the normalized graph Laplacian matrices of $\mathbf{A}$ and $\tilde{\mathbf{A}}$, and learn the eigenfunctions associated with the smallest K eigenvalues. However, the NeuralEF approach instead deploys neural approximations to the *principal* eigenfunctions, which correspond to the largest eigenvalues, of a kernel [12]. To this end, we move our focus from the normalized Laplacian matrix to the normalized adjacency matrix whose K principal eigenfunctions exactly correspond to the eigenfunctions associated with the K smallest eigenvalues of the former. Specifically, let $\mathbf{D} := \text{diag}(\mathbf{A}\mathbf{1}), \tilde{\mathbf{D}} := \text{diag}(\tilde{\mathbf{A}}\mathbf{1})$ denote the degree matrices. We then define the kernel function κ using the weighted sum of the normalized adjacency matrices, detailed below:

$$\kappa : \kappa(\mathbf{F}, \mathbf{F}) = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} + \alpha \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}, \tag{4}$$

where α is a trade-off parameter. Here we define the kernel in the space of features extracted from pre-trained models while other choices are also viable: it only affects the input of neural eigenfunctions. In the current setting, the features from pre-trained models are directly fed into neural eigenfunctions, which is sensible thanks to the great potential of pre-trained models and can prevent unnecessary training costs.

3.4. Learning Neural Eigenfunctions

Let $\mathbf{f} \in \mathbb{R}^C$ denote a row vector of $\mathbf{F}$ and $p(\mathbf{f})$ a uniform distribution over $\{\mathbf{F}_i\}_{i=1}^{NHW/P^2}$. The NeuralEF technique [12] approximates the K principal eigenfunctions of the kernel κ w.r.t. $p(\mathbf{f})$ with a K-output neural function $\psi : \mathbb{R}^C \to \mathbb{R}^K$ by solving the following problem:

$$\max_{\psi} \sum_{j=1}^{K} \mathbf{R}_{j,j} - \beta \sum_{j=1}^{K} \sum_{i=1}^{j-1} \widehat{\mathbf{R}}_{i,j}^2, \ s.t. \ \mathbb{E}_{p(\mathbf{f})}[\psi(\mathbf{f}) \circ \psi(\mathbf{f})] = \mathbf{1}, \tag{5}$$

where β is a positive coefficient and $\circ$ is Hadamard product, and

$$\begin{aligned} \mathbf{R} &:= \mathbb{E}_{p(\mathbf{f})}\mathbb{E}_{p(\mathbf{f}')}[\kappa(\mathbf{f}, \mathbf{f}')\psi(\mathbf{f})\psi(\mathbf{f}')^\top] \\ \widehat{\mathbf{R}} &:= \mathbb{E}_{p(\mathbf{f})}\mathbb{E}_{p(\mathbf{f}')}[\kappa(\mathbf{f}, \mathbf{f}')\hat{\psi}(\mathbf{f})\psi(\mathbf{f}')^\top]. \end{aligned} \tag{6}$$

Here $\hat{\psi}$ is the non-optimizable variant of ψ.

Adapting such results to our case and applying Monte Carlo estimation to the expectation yields

$$\mathbf{R} \approx \frac{1}{B^2}\mathbf{\Psi} \cdot \kappa(\mathbf{F}^B, \mathbf{F}^B) \cdot \mathbf{\Psi}^\top \text{ and } \widehat{\mathbf{R}} \approx \frac{1}{B^2}\widehat{\mathbf{\Psi}} \cdot \kappa(\mathbf{F}^B, \mathbf{F}^B) \cdot \mathbf{\Psi}^\top, \tag{7}$$

where $\mathbf{F}^B := [\mathbf{f}_1, \dots, \mathbf{f}_B]^\top \in \mathbb{R}^{B \times C}$ is a mini-batch of features, $\mathbf{\Psi} := [\psi(\mathbf{f}_1), \dots, \psi(\mathbf{f}_B)] \in \mathbb{R}^{K \times B}$ is the collection of the corresponding NN outputs, and $\widehat{\mathbf{\Psi}} := \text{stop\_gradient}(\mathbf{\Psi})$. The stop_gradient operation can be easily found in auto-diff libraries and plays a vital role in establishing the asymmetry between the K principal eigenfunctions. We put an L^2-batch normalization layer [12] at the end of model ψ to enforce the constraint in Equation (5).

ψ can be implemented as a neural network composed of convolutions or transformer blocks, which takes dense image features $\boldsymbol{f} = f(\boldsymbol{x}) \in \mathbb{R}^{H/P \times W/P \times C}$ as input and outputs $\psi(\boldsymbol{f}) \in \mathbb{R}^{H/P \times W/P \times K}$. After training, it can be used to predict new test data without relying on expensive test-time spectral decomposition.

3.5. Quantizing Neural Eigenfunctions

After obtaining the spectral embedding $\psi(f(x))$ for image patches, we should, by convention, invoke a Euclidean clustering algorithm such as K-means [29] to get clustering assignments. In practice, datasets can often be large, so it would be better to leverage online clustering mechanisms to avoid unaffordable storage costs. However, this approach may still be time-consuming when aiming for good convergence. Additionally, this pipeline is not as flexible as NN-based segmenters.

To bridge the gap, we impose a constraint on the output of ψ to be K-dim one-hot vectors which directly indicate clustering assignments. We resort to the Gumbel-Softmax estimator [20] for gradient-based optimization. This follows the notion that the outputs of eigenfunctions are soft clustering assignments, and we further perform quantization of them. This is also in a similar spirit to the spectral hashing approach [43] where the output of ψ is assumed to be vectors over $\{-1, 1\}^K$. We clarify that the Gumbel-Softmax estimator precedes the aforementioned L^2 batch normalization layer. In the test phase, we remove this estimator and the L^2 batch normalization layer, using the NN outputs directly as softmax logits for clustering. This results in a pure NN-based workflow for spectral clustering.

3.6. From Clusters to Image Segments

During inference, we up-sample the softmax logits bilinearly to match the original resolution of input images. We then apply the argmax operation to obtain discrete clustering assignments. A natural question that arises is how to match these clustering assignments to pre-defined semantics to obtain the final image segments. Two well-studied solutions are Hungarian matching [27] and majority voting. As per DSM [30], when evaluated on standard image semantic segmentation benchmarks, we first collect the clustering assignments for all validation images and then match them to ground-truth labels via these approaches to conduct a quantitative evaluation of segmentation performance.

4. Experiments

To evaluate the efficacy of the proposed method for unsupervised semantic segmentation, we conduct comprehensive experiments on various standard benchmarks.

4.1. Experimental Setups

Datasets. We primarily experiment on Pascal Context and Cityscapes, which consist of 60 and 27 classes, respectively. We employ the same data pre-processing and augmentation strategies as the work [40] on the training dataset.

Pre-trained models. We use pre-trained ViTs [14] provided by the timm library [44] and consider the "Small", "Base", and "Large" variants. The weights have been pre-trained on ImageNet-21k [36] and fine-tuned on ImageNet [10] in a supervised manner. In particular, we consider pre-trained models at a high resolution, like 384×384, to obtain fine-grained segments. We fix the patch size to 16×16 (i.e., $P = 16$) to better trade-off efficiency and accuracy. We use the output of the last attention block of ViTs as $\boldsymbol{f}_i$ without elaborate selection. We also feed the intermediate features of the pre-trained ViTs, which can be freely accessed, to the neural eigenfunctions to enrich the input information.

Modeling and training. We set $k = 256$ for the nearest neighbor graph defined on pre-trained models' features. To reduce the cost of searching for the nearest neighbors, we confine the search to the current mini-batch, which is shown to be empirically effective. We specify the other graph following DSM [30]. The trade-off coefficient α equals 0.3 based on an ablation study reported in Section 4.4. The training objective is detailed in Equation (5) and unfolded in Equation (7). As K implies the number of semantic classes uncovered automatically, we make it larger than the number of ground-truth semantic classes and, in practice, set it to 256. We set the trade-off coefficient β to 0.08 and linearly scale it for other values of K (see the study in Section 4.4). We use 2 transformer blocks with linear self-attention [22] and a linear head to specify the neural eigenfunctions ψ for efficiency. We restrict the weight matrix of the linear head to have orthogonal columns and find it beneficial to the final performance empirically. We anneal the temperature for Gumbel-Softmax from 1 to 0.3 following a cosine schedule during training. The training relies on an Adam optimizer [25] and a learning rate of 10^{-3} (with cosine decay). No weight decay is employed. The training lasts for 40 epochs with batch size 16 (or 8 if an out-of-memory error occurs), which takes half a day on one RTX 3090 GPU.

Evaluation. We test on the validation set of the datasets and report both pixel accuracy (Acc.) and mean intersection-over-union (mIoU). We follow the standard practice in unsupervised semantic segmentation where the validation images are resized and cropped to have 320×320 pixels (see [16, 39]). In particular, the background class in the Pascal Context is excluded from evaluation. We apply a softmax operation to the outputs of ψ to obtain clustering assignments. We then use majority voting to match the resulting clusters to semantic segments. We apply CRF [26] for post-processing (although it is empirically shown that its contribution to performance gain is marginal).

4.2. Comparison with Leading Methods

We first compare the proposed method to leading unsupervised semantic segmentation methods. We consider two representative competitors from the literature: MaskCLIP [48] and ReCo [39], which empirical outperform a variaty of baselines such as IIC [21], PiCIE [7], and STEGO [16]. In particular, MaskCLIP uses the ViT-B backbone trained by CLIP [35], whereas ReCo uses the ViT-L/14@336px, also trained by CLIP. We also try to include the self-training variants of MaskCLIP and ReCo in com-

Method	**Acc.** (%)	**mIoU** (%)
MaskCLIP [48]	-	25.5
MaskCLIP+ [48]	-	31.1
ReCo [39]	51.6	27.2
K-means		
ViT-S	61.9	28.9
ViT-B	58.7	30.9
ViT-L	45.3	19.3
Ours		
ViT-S	70.4	38.8
ViT-B	69.7	37.5
ViT-L	63.2	33.2
*Ours**		
ViT-S	**74.6**	**39.6**
ViT-B	73.2	37.6
ViT-L	71.9	35.0

Table 1. Comparisons on unsupervised semantic segmentation performance on Pascal Context [31]. The results of MaskCLIP and ReCo are from the original papers.

Method	**Acc.** (%)	**mIoU** (%)
MaskCLIP [48]	35.9	10.0
ReCo [39]	74.6	19.3
ReCo+ [39]	83.7	24.2
K-means		
ViT-S	77.0	22.4
ViT-B	74.8	23.2
ViT-L	66.3	20.9
Ours		
ViT-S	83.4	28.2
ViT-B	81.4	26.8
ViT-L	80.3	26.3
*Ours**		
ViT-S	**84.6**	30.0
ViT-B	84.2	**30.7**
ViT-L	84.2	30.0

Table 2. Comparisons on unsupervised semantic segmentation performance on Cityscapes [9]. The results of MaskCLIP and ReCo are from the original papers.

parison. Besides, we introduce two other baselines: (*i*) fit a K-means with the features of pre-trained models for training data (using the same number of clustering centers as our method) and use it to predict clustering assignments for validation data; (*ii*) likewise, fit a K-means with the features preceding the linear head in ψ. The two ways are referred to as "K-means" and "Ours*" in our studies. We have not compared with DSM [30] because the involved divide-and-conquer procedure for synchronizing the clustering results across different images is non-trivial to implement. In theory, DSM can lead to a similar segmentation performance to our method (yet with more resource consumption).

We report the results in Table 1 and Table 2. As shown, our methods outperform MaskCLIP, ReCo, and K-means with significant margins, which reflects the efficacy of the learned neural eigenfunctions for unsupervised semantic segmentation. "Ours*" even outperforms "Ours", which is probably attributed to the fact that the features preceding the linear head in ψ have a much higher dimension than the final outputs and thus are more informative. While MaskCLIP+ and ReCo+ employ an extra time-consuming self-training step, they are still inferior to our methods. It is reasonable to speculate that combining our methods with self-training can further improve performance. Note also that K-means can serve as a strong baseline for unsupervised semantic segmentation. This finding appears to contradict the results reported in DSM [30] (Table 4). We deduce the reason that DSM sets the number of clusters to the number of true classes rather than a larger quantity and performs K-means directly on the validation set instead of the training set. Besides, as the size of the pre-trained model increases, both K-means and our methods yield worse unsupervised segmentation performance. We attribute this to larger pre-trained models producing more abstract outputs that contain more overall semantics than specialized details. In contrast, smaller pre-trained models may fall short in expressiveness. Therefore, we suggest using a pre-trained model with appropriate capacity when applying the proposed method to new applications.

Qualitative results. Figure 2 shows some qualitative results of the proposed methods on Pascal Context. We also include the K-means baseline in comparison. Most notably, unlike MaskContrast [41], our method can detect multiple semantic categories in the same image, and the generated object boundaries are sharp. Furthermore, our spectral clustering-based approach can greatly reduce the chaotic fragments produced by K-means. We defer the visualization for the learned neural eigenfunctions and the segmentation results on Cityscapes to Appendix.

4.3. Evaluation in More Scenarios

In this subsection, we evaluate the proposed method in more scenarios to study its behavior systematically.

First, we consider the evaluation protocol popularly used in the *supervised* setting. Specifically, following the setup [40], we leverage a sliding window with the same resolution as training images to cope with the varying sizes of evaluation images. The background class in the Pascal Context is included for evaluation, and the results on Cityscapes cover only 19 categories. Due to implementation challenges, we have not included existing methods in comparison, but we introduce a supervised semantic segmentation baseline, DeepLabv3+ [5], which can serve as an upper bound on performance.

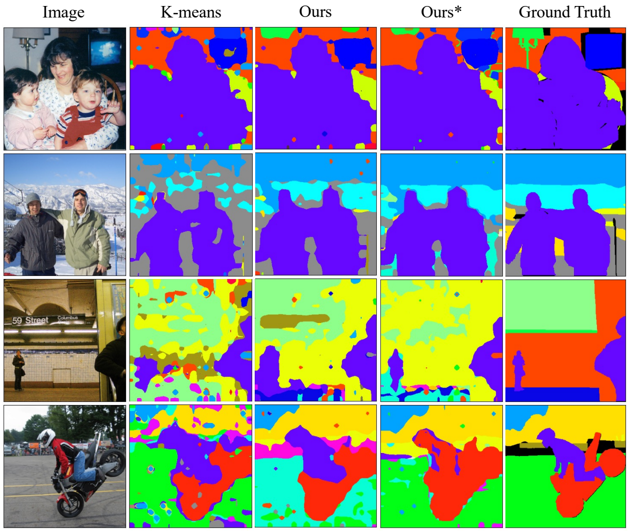

Figure 2. Visualization of the unsupervised semantic segmentation results on Pascal Context [31].

Method	**Acc.** (%)	**mIoU** (%)
Pascal Context		
Supervised	-	48.5
K-means	56.3	31.9
Ours	67.4	**41.4**
*Ours**	**68.9**	41.3
Cityscapes		
Supervised	-	77.3
K-means	75.5	34.2
Ours	86.1	46.7
*Ours**	**88.3**	**52.8**

Table 3. Comparisons on semantic segmentation performance using the widely adopted evaluation protocol [40]. In particular, the background class in the Pascal Context is included for evaluation, and the results on Cityscapes cover 19 categories. The pre-trained models with ViT-S architecture are used. Supervised results for the two datasets rely on DeepLabv3+ [5] using ResNet-101 [17] and Xception-65 [8] backbones, respectively.

As shown in Table 3, in this new evaluation setting, our methods can clearly outperform K-means, and "Ours*" is slightly better than "Ours" in general. These results are consistent with those reported in Section 4.2 and help to verify the extensibility of the proposed method. We also notice that the performance gap between the proposed *unsupervised* segmentation method and the *supervised* baseline is not significant, especially on the Pascal Context dataset, which motivates further investigation on the direction of improving spectral clustering.

Method	**Acc.** (%)	**mIoU** (%)
K-means	50.7	19.2
Ours	**63.3**	21.6
*Ours**	62.5	**23.6**

Table 4. Comparisons on unsupervised semantic segmentation performance on ADE20K using the evaluation protocol of [40]. The pre-trained model with ViT-S architecture is used.

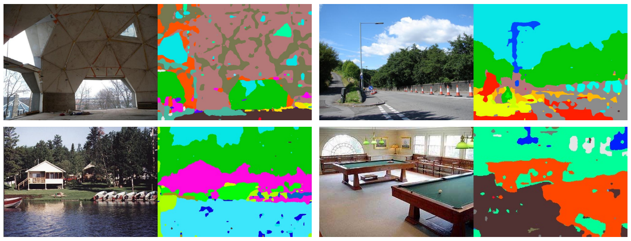

Figure 3. Visualization of the unsupervised semantic segmentation results of our method on ADE20K. In each pair, the left refers to the input image, and the right refers to the segmentation result. As shown, our method can yield reasonable pixel groups for images containing complex structures.

After that, we assess our method on ADE20K, one of the most challenging semantic segmentation datasets that contain 150 fine-grained semantic categories. To tackle a large number of ground-truth semantic classes, we set K to 512 with β as 0.04. The training lasts for 20 epochs, given that the training set is relatively large. We use the evaluation protocol of [40]. We include the K-means baseline based on our own implementation for a fair comparison.

The results are displayed in Table 4. As shown, our methods are still superior to the K-means baseline, especially regarding pixel accuracy. The best mIoU is 23.6%, much lower than the corresponding results on Pascal Context and Cityscapes. This probably stems from the large number of semantic categories in ADE20K—the used pre-trained models are not trained with fine-grained objectives (e.g., losses defined on patches), so they cannot ensure the semantic richness of the extracted dense features, which makes our method tend to generate clusters that conjoin multiple fine-grained semantic categories (see the lamp and the streetlight in Figure 3). One potential solution to this problem is fine-tuning the pre-trained models with patch-wise self-supervised learning loss and then invoking the proposed spectral clustering pipeline.

4.4. Ablation Studies

In this subsection, we present a comprehensive study of several key hyper-parameters in our method. We evaluate using the same setting as in Section 4.2. We consider the ViT-S architecture for per-trained models given its superior performance testified in previous results.

The output dimension K. K determines the number of eigenfunctions to learn and hence the dimension of the spectral embedding. It also implicitly connects to the number of

K	64	128	256	512
Acc. (%)	67.1	71.6	70.4	71.1
mIoU (%)	27.8	33.2	38.8	37.9

Table 5. Performance comparison of different K based on a pre-trained ViT-S model on Pascal Context validation set.

α	0	0.1	0.3	0.5	0.7
Pascal Context	38.6	37.9	38.8	38.9	38.0
Cityscapes	27.2	26.7	28.2	28.1	27.9

Table 6. Mean IoU (%) comparison of different trade-off coefficients α based on a pre-trained ViT-S model.

semantic classes uncovered automatically during training. In previous experiments, we keep K larger than the number of ground-truth semantic classes. Here we perform an ablation study on K on Pascal Context to reflect the necessity of doing so. The results are summarized in Table 5, which indicates that a value of K of at least 256 is necessary to achieve superior performance in terms of mIoU scores. A larger K yields more expressive representations that capture subtleties yet at the cost of increased computational overhead. Moreover, it should be noted that the NeuralEF technique may fail to uncover the eigenfunctions with small eigenvalues [12]. Therefore, we suggest selecting a moderate value of K in practice.

The trade-off parameter α. The proposed spectral clustering workflow is compatible with any kernel function that captures plausible relationships between image patches. Currently, the used kernel is a weighted sum of the normalized adjacency defined on features from pre-trained models and that defined on down-sampled pixels. To verify the robustness of our method to the trade-off parameter α, we perform an ablation study on it and report the results in Table 6. We include results for both Pascal Context and Cityscapes for a thorough investigation. As shown, the mIoU on the validation data does not vary significantly w.r.t. α, and limiting α to $[0.3, 0.5]$ can lead to superior results. Note that when $\alpha = 0$, i.e., we only use the features from pre-trained models to construct the graph kernel, the mIoU drops slightly, indicating that the features from pre-trained models can retain most information on the neighborhood relationship of raw pixels. This also presents an opportunity for enhancing performance by improving the graph defined on down-sampled image pixels.

The trade-off parameter β. The trade-off parameter β in Equation (5) for learning neural eigenfunctions affects the empirical convergence. To investigate whether our method is sensitive to the choice of β, in Table 7 we ablate the influence of β in both "Ours" and "Ours*" approaches on the Pascal Context data. We observe that the validation mIoU remains stable across β ranging from 0.04 to 0.16, which confirms the robustness of our method against β. With this, we set β to 0.08 in all experiments.

β	0.04	0.08	0.16
Ours	38.8	38.8	39.0
*Ours**	39.0	39.6	39.3

Table 7. Mean IoU (%) comparison of different β based on a pre-trained ViT-S model on Pascal Context validation set.

	Acc. (%)	mIoU (%)
Pascal Context	55.8	15.2
Cityscapes	81.2	18.5

Table 8. Zero-shot transfer results of our method. The training is performed on ImageNet based on the pre-trained ViT-S model.

Zero-shot transfer. Due to its unsupervised clustering nature, our method does not necessarily rely on target images for training. In this spirit, we perform an initial study where the training is conducted on ImageNet, but the evaluation is conducted on both Pascal Context and Cityscapes. This forms a zero-shot transfer paradigm for unsupervised semantic segmentation. The training lasts for 5 epochs under the same setting in previous studies. We report the results in Table 8. As a reference, the corresponding mIoU of MaskCLIP and ReCo on Cityscapes are 10.0 and 19.3 respectively [39], thus our method provides competitive performance. Nonetheless, the mIoUs for all the methods are much worse than those in previous studies. This is probably because images from ImageNet mostly contain clear foregrounds and single objects, which is in sharp contrast to the complex scene images in Pascal Context and Cityscapes, thus the learned neural eigenfunctions struggle to generalize. A potential remedy for this problem is training on more realistic datasets to reduce the transfer gap.

Majority voting vs. Hungarian matching. By having the same number of clusters as the ground truth classes and utilizing Hungarian matching, we can maintain a relatively low cluster count and probably enable the assignment of clusters to semantic classes by an expert in practice. We conduct a study on Pascal Context, where the number of clusters is set to match the semantic categories. The resulting mIoU of Hungarian matching is 0.235, slightly lower than the mIoU obtained with majority voting in the same setting (0.277). One possible explanation is that the semantic categories in the benchmarks are often structured, such as a car consisting of the body and the wheels. As a result, the automatically discovered clusters may not correspond one-to-one with the semantic categories. This observation also highlights the need for a larger number of clusters to address this issue effectively.

The pre-training method. When using ViT backbones pre-trained on ImageNet classification, we observe that stronger

ViT-B	ViT-L	ViT-L/14@336px
37.2	38.7	44.0

Table 9. Mean IoU (%) of our method on Pascal Context when using various backbones pre-trained by CLIP.

pre-trained models yield worse segmentation performance. We have realized backbones pre-trained by CLIP may exhibit a different tendency because the model outputs are coerced to conform with text supervision rather than being discriminative in recognizing foreground objects. We empirically study this in Table 9. As shown, the backbone pre-trained by CLIP exhibits an increasing trend on mIoU as the model expressiveness increases.

5. Conclusion

This work establishes an end-to-end NN-based pipeline for spectral clustering for unsupervised semantic segmentation. To achieve that, we build a connectivity graph over image patches using information from both pre-trained models and raw pixels and employ neural eigenfunctions to produce spectral embeddings corresponding to suitable graph kernels. We further quantize the output of the neural eigenfunctions to obtain clustering assignments without resorting to an explicit grouping step. After training, our method can generalize to novel test data easily and reliably. The reliance on pre-trained models gives our method good training efficiency and sufficient expressiveness. Extensive results confirm its superior performance over competing baselines.

One limitation is that, like most clustering-based methods, our method needs to be exposed to ground-truth semantic masks to match clustering assignments to semantic segments. Introducing text prompts to guide clustering is a potential solution and deserves future investigation.

Acknowledgments

We thank Bernhard Schölkopf for providing constructive suggestions and proofreading the paper. This work was supported by Natural Science Foundation of Shanghai (No. 23ZR1428700).

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Labels4free: Unsupervised segmentation using stylegan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13970–13979, 2021.

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

[3] Yoshua Bengio, Pascal Vincent, Jean-François Paiement, O Delalleau, M Ouimet, and N LeRoux. Learning eigenfunctions of similarity: linking spectral clustering and kernel pca. Technical report, Technical Report 1232, Departement d'Informatique et Recherche Oprationnelle ..., 2003.

[4] Lyndon Chan, Mahdi S Hosseini, and Konstantinos N Plataniotis. A comprehensive analysis of weakly-supervised semantic segmentation in different image domains. *International Journal of Computer Vision*, 129:361–384, 2021.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. *Advances in neural information processing systems*, 32, 2019.

[7] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16794–16804, 2021.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Zhijie Deng, Jiaxin Shi, Hao Zhang, Peng Cui, Cewu Lu, and Jun Zhu. Neural eigenfunctions are structured representation learners. *arXiv preprint arXiv:2210.12637*, 2022.

[12] Zhijie Deng, Jiaxin Shi, and Jun Zhu. Neuralef: Deconstructing kernels by deep neural networks. *arXiv preprint arXiv:2205.00165*, 2022.

[13] William E Donath and Alan J Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[16] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7014–7023, 2018.

[19] 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/CVF International Conference on Computer Vision*, pages 7334–7344, 2019.

[20] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[21] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.

[22] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[23] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. *arXiv preprint arXiv:2105.00957*, 2021.

[24] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017.

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

[26] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

[27] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[28] Marc T Law, Raquel Urtasun, and Richard S Zemel. Deep spectral clustering learning. In *International conference on machine learning*, pages 1985–1994. PMLR, 2017.

[29] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[30] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022.

[31] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014.

[32] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[33] Yassine Ouali, Céline Hudelot, and Myriam Tami. Autoregressive unsupervised image segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 142–158. Springer, 2020.

[34] David Pfau, Stig Petersen, Ashish Agarwal, David GT Barrett, and Kimberly L Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. *arXiv preprint arXiv:1806.02215*, 2018.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[36] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021.

[37] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks—ICANN'97: 7th International Conference Lausanne, Switzerland, October 8–10, 1997 Proceeedings*, pages 583–588. Springer, 2005.

[38] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[39] Gyungin Shin, Weidi Xie, and Samuel Albanie. Reco: Retrieve and co-segment for zero-shot transfer. *arXiv preprint arXiv:2206.07045*, 2022.

[40] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[41] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10052–10062, 2021.

[42] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Object segmentation without labels with large-scale generative models. In *International Conference on Machine Learning*, pages 10596–10606. PMLR, 2021.

[43] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *Advances in neural information processing systems*, 21, 2008.

[44] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[45] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4066–4075, 2019.

[46] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. *Advances*

in Neural Information Processing Systems, 33:16579–16590, 2020.

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

[48] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022.