Deciphering ‘What’ and ‘Where’ Visual Pathways from Spectral Clustering of Layer-Distributed Neural Representations

Xiao Zhang*
University of Chicago
zhang7@uchicago.edu

David Yunis*
TTI-Chicago
dyunis@ttic.edu

Michael Maire
University of Chicago
mmaire@uchicago.edu

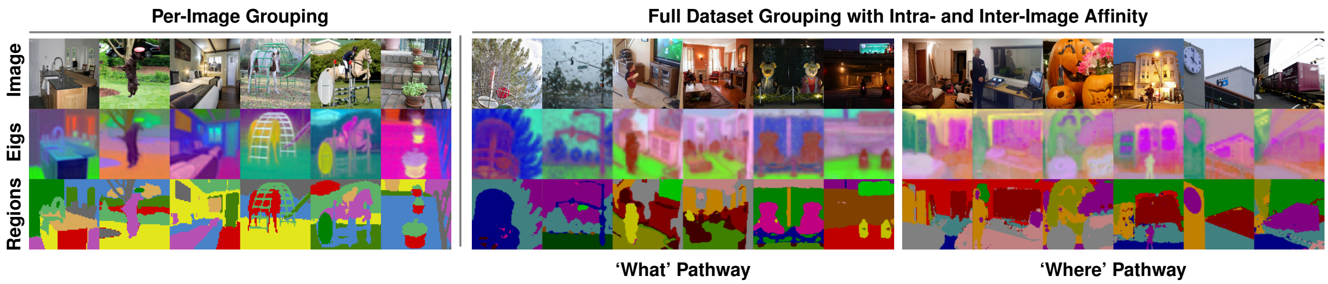


Figure 1. Our novel optimization procedure, resembling spectral clustering, leverages features throughout layers of a pre-trained model to extract dense structural representations of images. Shown are results of applying our method to Stable Diffusion [53]. ***Left:*** Analyzing internal feature affinity for a single input image yields region grouping. ***Right:*** Extending the affinity graph across images yields coherent dataset-level segmentation and reveals ‘what’ (object identity) and ‘where’ (spatial location) pathways, depending on the feature source.

Abstract

We present an approach for analyzing grouping information contained within a neural network’s activations, permitting extraction of spatial layout and semantic segmentation from the behavior of large pre-trained vision models. Unlike prior work, our method conducts a wholistic analysis of a network’s activation state, leveraging features from all layers and obviating the need to guess which part of the model contains relevant information. Motivated by classic spectral clustering, we formulate this analysis in terms of an optimization objective involving a set of affinity matrices, each formed by comparing features within a different layer. Solving this optimization problem using gradient descent allows our technique to scale from single images to dataset-level analysis, including, in the latter, both intra- and inter-image relationships. Analyzing a pre-trained generative transformer provides insight into the computational strategy learned by such models. Equating affinity with key-query similarity across attention layers yields eigenvectors encoding scene spatial layout, whereas defining affinity by value vector similarity yields eigenvectors encoding object identity. This result suggests that key and query vectors coordinate attentional information flow according to spatial proximity (a ‘where’ pathway), while value vectors refine a semantic category representation (a ‘what’ pathway).

*Equal contribution. Code available at this link.

1. Introduction

An explosion in self-supervised learning techniques, including adversarial [23, 31, 32], contrastive [11, 12, 26, 72], reconstructive [34, 66], and denoising [29, 60] approaches, combined with the focus on training large-scale foundation models [4] on vast collections of image data has produced deep neural networks exhibiting dramatic new capabilities. Recent examples of such models include CLIP [51], DINO [8], MAE [27], and Stable Diffusion [53]. As training is no longer primarily driven by annotated data, there is a critical need to understand what these models have learned, provide interpretable insight into how they work, and develop techniques for porting their learned representations for use in accomplishing additional tasks.

However, interpretable analysis of neural networks is challenging. Procedures such as guided backpropagation [61] or Grad-CAM [57] assist with interpretability with respect to particular labels, but are limited in scope. Others propose heuristics for extracting information relevant to particular downstream tasks, or analyze specific features in models [2, 9, 10, 13, 19, 28, 39, 41, 63, 77]. The distributed nature of both the information encoded within deep networks [62] and their computational structure frustrates the development of general-purpose techniques.

It is similarly unclear how best to repurpose pre-trained models toward downstream tasks. Task-specific heuristics, fine-tuning on labeled data, prompt engineering (if applicable), or clustering frozen feature representations might all

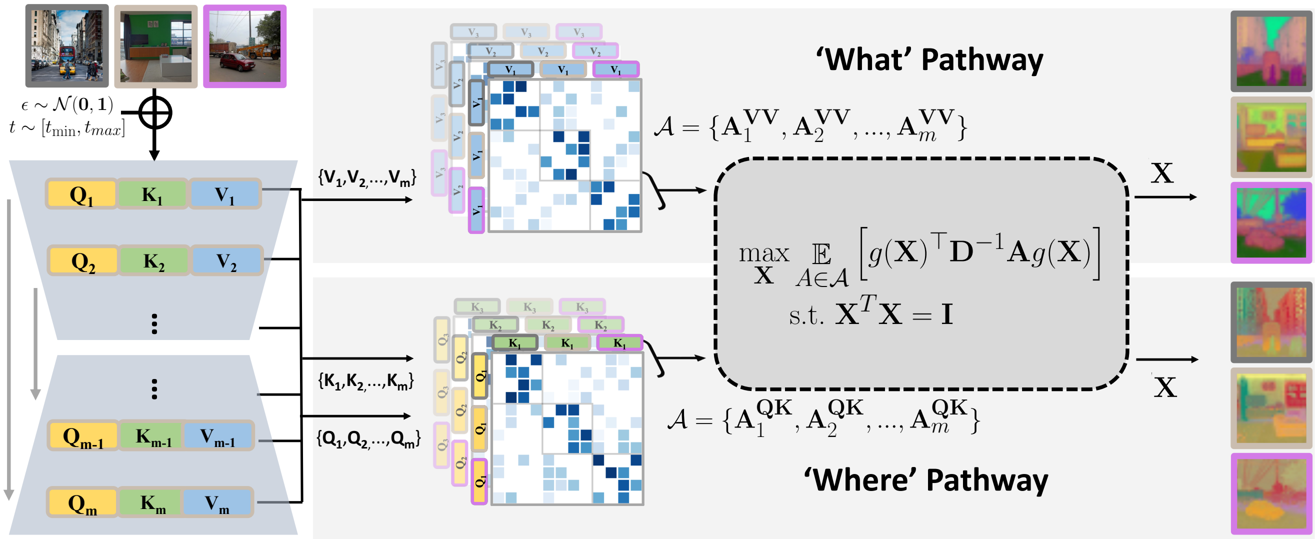


Figure 2. **Spectral clustering of layer-distributed representations.** For each input image, we collect key, query, and value feature vectors from attention layers across network depth (and, for diffusion models, time). Intra- and inter-image value-value (top) and key-query (bottom) similarity define a collection of affinity matrices indexed by layer (and time). We solve for pseudo-eigenvectors $\boldsymbol{X}$ which, when scaled to the spatial resolution of each layer via $g(\cdot)$, best satisfy an average of per-layer spectral partitioning criteria. The leading eigenvector from value-value affinity reveals semantic category *(top)*, while that from key-query affinity reveals spatial layout *(bottom)*.

be viable options. Yet, an element of art remains in choosing which features to extract or which layers to fine-tune.

We introduce a new analysis approach that provides insight into model function and directly extracts significant visual information about image segmentation, as shown in Figure 1, with neither a-priori knowledge of, nor hyperparameter search over, where such information is stored in the network. We accomplish this through an analysis that couples the entire activation state of the network, from shallow to deep layers, into a global spectral clustering objective. Solving this clustering problem not only yields new feature representations (in the form of eigenvector embeddings) directly relevant to downstream segmentation tasks, but also, as Figure 2 illustrates, provides insight into the inner workings of vision models. Our contributions include:

- A new approach, inspired by spectral clustering, for wholistic analysis of deep neural network activations.
- Improved quality of extracted regions across models, compared to variants analyzing single layers.
- An efficient gradient-based optimization framework that enables our approach to scale to joint analysis of network behavior across an entire dataset simultaneously.
- Unsupervised semantic segmentation results on par with STEGO [24], but extracted from a pre-trained generative model rather than a contrastive backbone.
- Insight into the computational strategy learned by large-scale vision models: internal features are partitioned into 'what' and 'where' pathways, which separately maintain semantic and spatial information.

2. Related Work

Image segmentation. Segmentation, as a generic grouping process, has historically been regarded as an important intermediate task in computer vision. Significant efforts focus on building object-agnostic methods for partitioning an image into coherent regions or, equivalently, their dual representation as contours [1, 3, 6, 18, 35, 52], with standard benchmarks [45] driving progress. Semantic and instance segmentation, which aim to extract image regions corresponding to specific category labels or object instances, have undergone parallel development, driven by benchmark datasets such as PASCAL [21] and COCO [40]. Notable modern supervised methods utilize CNN [25] or Transformer [7] architectures trained in an end-to-end fashion. Particularly relevant is recent work demonstrating the ability of models to learn to segment with relatively few labels [2, 78]. Spectral clustering, as a method of approximating the solution of a graph partitioning objective [59], has appeared as a core algorithmic component across a variety of segmentation systems [1, 35, 43, 44, 59, 64, 75, 76].

Segmentation without labels. Recent methods, such as DINO [8], learn intra-image and inter-image correspondences between pixels without the need for dense labels. STEGO [24] and PiCIE [14] propose to cluster pixel-wise features of a self-supervised backbone, showing impressive performance on semantic segmentation with no labels at all. LSeg [38] and GroupViT [73] modify CLIP [51] to enable zero-shot open-vocabulary semantic segmentation.

Another class of methods builds entirely on top of existing models, with no additional training [19, 39, 41, 77]. Recent attempts at instance segmentation [69, 71] yield impressive results through heuristic decoding strategies based on the structure of a particular model's features (*e.g.,* the final layer of DINO [8]). Other work, based on Stable Diffusion [53], finds unsupervised dense correspondences using the text embedding space as a shared anchor [28] or through careful choice of features [63].

Interpretability. Grad-CAM [57], layer-wise relevance propagation [49], and guided backpropagation [61] provide heuristics to visualize the responsibility of different input spatial regions for predictions of a deep neural network. Other approaches visualize attention matrices to find salient input regions for NLP [15, 36] and vision [9, 10, 74] tasks. Chen *et al.* [13] find evidence of depth information inside Stable Diffusion. Yet, visualizing and interpreting neural network behavior remains a challenging problem due to the distributed nature of the representations they learn [62].

Spectral clustering of neural features. TokenCut [71], MaskCut [70], and DSM [47] define affinity graphs using final features of a pre-trained DINO [8] model, and use spectral clustering to segment the original image. We take inspiration from these approaches and utilize them as baselines for experimental comparison. Our methodology differs in being global and accounting for features throughout the network, rather than restricted to one layer.

Neuroscience perspectives on visual processing streams. Trevarthen [65] and Schneider [56] propose the concept of separate visual processing pathways in the brain for localization ('where') and discrimination ('what'). Mishkin *et al.* [48] review evidence for this specialization of processing in the monkey, while subsequent work examines specialized pathways in terms of perception and action [22], as well as spatial memory and navigation [37]. While these ideas motivate our investigation into information stored in the key, query, and value vectors distributed throughout a Transformer architecture, the question of relevance (if any) to biological vision systems is beyond our scope.

3. Method

Our method closely resembles spectral clustering applied simultaneously across attention layers within a given neural network. The following sections detail our full method and discuss different graph construction choices for spectral clustering, which respectively allow us to extract different kinds of information from source models.

3.1. Spectral Clustering with Distributed Features

Spectral clustering formulates the data grouping problem from the view of graph partitioning. It uses the eigenvectors of the normalized Laplacian matrix to partition the data into balanced subgraphs with minimal cost of breaking edges [59]. Specifically, with a symmetric affinity matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, where N denotes the total number of data points and entries $\boldsymbol{A}_{ij} \geq 0$ measure the similarity between data samples with indices i and j, we can embed the data into a lower dimensional representation $\boldsymbol{X} \in \mathbb{R}^{N \times C}$, where C denotes the number of feature channels and $C \ll N$, as the solution of the following generalized eigenproblem:

$$(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{X} = \lambda \boldsymbol{D}\boldsymbol{X}. \tag{1}$$

$\boldsymbol{D}$ is the diagonal degree matrix of $\boldsymbol{A}$ with diagonal entries $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$, and $\boldsymbol{X}, \lambda$ are eigenvectors and eigenvalues respectively. We can then produce a discrete partition from $\boldsymbol{X}$ through K-Means clustering.

Though spectral clustering is a powerful tool for data analysis, its performance is highly dependent on the choice of affinity matrix. Recent works [47, 70, 71] apply spectral clustering on an affinity matrix constructed from features in the last layer of DINO [8], yielding strong performance in segmentation tasks. However, the choice of graph may not be clear when the desired information is distributed across the layers of a neural network, or noise levels in diffusion models [2]. Therefore, we extend Eqn. 1 to allow for constructing $\boldsymbol{A}$ using multiple sources of information. A classic approach to solve Eqn. 1 with a set of affinity matrices, $\mathcal{A} = \{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m\}$, is the constrained spectral clustering problem [17]. It constructs a block diagonal affinity matrix from this set:

$$\boldsymbol{A}_{\mathcal{A}} = \begin{bmatrix} \boldsymbol{A}_1 & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{A}_m \end{bmatrix} \tag{2}$$

and imposes additional cross-scale consistency constraints. However, the size of this matrix, and the computational expense of solving the resulting eigenproblem, can become intractable with increasing $|\mathcal{A}|$. Instead of solving the original eigenproblem in Eqn. 1, we solve an approximation:

$$\begin{aligned} &\max_{\boldsymbol{X}} \mathop{\mathbb{E}}_{\boldsymbol{A} \in \mathcal{A}} \left[g(\boldsymbol{X})^\top \boldsymbol{D}_{\boldsymbol{A}}^{-1} \boldsymbol{A} g(\boldsymbol{X}) \right], \\ &\text{s.t. } \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I} \end{aligned} \tag{3}$$

where $g(\cdot)$ corresponds to the resampling function that bilinearly interpolates the spatial resolution of $\boldsymbol{X}$ to match the size of $\boldsymbol{A}$, allowing affinity matrices to be constructed from feature maps with varying resolutions. In Eqn. 3, we follow Meila and Shi [46] to solve the spectral clustering from the random walk perspective, since the random walk matrix and the attention matrix have the same format and the same eigenvectors. This objective encodes a Rayleigh quotient optimization simultaneously across affinities in $\mathcal{A}$, which avoids the intractable exact solution and can naturally scale with increasing $|\mathcal{A}|$.

Notice that $\boldsymbol{D}_{\boldsymbol{A}}^{-1}\boldsymbol{A}$ is a random walk matrix with maximum eigenvalue of 1. For numerical stability, we impose a constraint to ensure that the maximum value of the objective does not exceed 1. In addition, we replace the strict orthogonality requirement with a soft Frobenius regularization term whose coefficient is 1. Consequently, our final optimization objective is:

$$\min_{\boldsymbol{X}} \mathop{\mathbb{E}}_{\boldsymbol{A}\in\mathcal{A}} |g(\boldsymbol{X})^\top \boldsymbol{D}_{\boldsymbol{A}}^{-1}\boldsymbol{A}g(\boldsymbol{X}) - 1| + \|\boldsymbol{X}^\top\boldsymbol{X} - \boldsymbol{I}\|_F. \quad (4)$$

We parameterize $\boldsymbol{X}$ as a learnable feature map and solve for it using gradient-based optimization. In the following subsections, we discuss the choice of affinity set $\mathcal{A}$ and how that choice affects the information we extract.

3.2. Per-Image Analysis

Attention layers in vision models naturally consider patchwise relationships when computing the attention matrix. We can use this matrix as an affinity graph for spectral clustering, which allows investigating how a model groups regions in an image internally, without imposing outside heuristics. For the Vision Transformer [20] and U-Net [54] variants that include a total of m attention blocks, we build an affinity set $\mathcal{A} = \{\boldsymbol{A}_l^{\boldsymbol{QK}}\}_{l=1}^m$ across layers, where $\boldsymbol{A}_l^{\boldsymbol{QK}}$ is the pre-softmax self-attention matrix [67] at layer l.

$$\boldsymbol{A}_l^{\boldsymbol{QK}} = \exp\left(\frac{\boldsymbol{Q}_l\boldsymbol{K}_l^\top}{\sqrt{d_l}}\right) \in \mathbb{R}^{N\times N}, \quad (5)$$

where $\boldsymbol{Q}_l, \boldsymbol{K}_l \in \mathbb{R}^{N\times d_l}$ are the query and key matrices for that layer respectively, and d_l is the embedding dimension.

3.3. Full-Dataset Extension

We can extend the self-attention operation in a single image to affinity matrix construction across different images. This allows probing how models relate different regions across different images using their internal computational structure. Specifically, we construct graphs similar to single-image self-attention matrices by computing normalized pairwise dot products between queries at every position in one image, and keys at every position in another. Scaling to large datasets, we extract one set of features $\boldsymbol{X}_i$ for each image with index i in the dataset. To do this, we optimize a mini-batch of features:

$$\boldsymbol{X}_{\text{batch}} = \begin{bmatrix} \boldsymbol{X}_j \\ \vdots \\ \boldsymbol{X}_k \end{bmatrix} \in \mathbb{R}^{(N\cdot B)\times C}, \quad (6)$$

and construct graphs over that mini-batch:

$$\boldsymbol{A}_l^{\boldsymbol{QK}} = \begin{bmatrix} \widehat{\boldsymbol{Q}}_{j,l} \\ \vdots \\ \widehat{\boldsymbol{Q}}_{k,l} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{K}}_{j,l}^\top \ldots \widehat{\boldsymbol{K}}_{k,l}^\top \end{bmatrix} \in \mathbb{R}^{(N\cdot B)\times(N\cdot B)}, \quad (7)$$

where $\widehat{\boldsymbol{Q}}_{j,l}, \widehat{\boldsymbol{K}}_{j,l} \in \mathbb{R}^{N\times d_l}$ represent the queries and keys for image j at layer l normalized to unit-norm, and there are B images in a mini-batch. We normalize vectors as calibrating magnitudes across images is not trivial.

Though we limit the graph to a mini-batch, it is still prohibitively expensive to store and optimize over. Thus, we sparsify the graph by only keeping the top c_{intra} intra-image connections and the top c_{inter} inter-image connections for each location. In addition, we set all values below a threshold to 0. To investigate what kind of information models mix across spatial locations, we consider a similar affinity set $\mathcal{A} = \{\boldsymbol{A}_l^{\boldsymbol{VV}}\}_{l=1}^m$ built from the value matrices $\widehat{\boldsymbol{V}}_{i,l}$.

With these approximate layer-wise graphs, we optimize the objective in Eqn. 4 a small number of steps per mini-batch, then sample a new mini-batch of images and continue. Finally, this process discovers a consistent set of dense features for a dataset. A visualization of the entire method can be found in Figure 2.

3.4. Recovering Orthogonal Representations

Eqn. 4 suggests an approximate formulation of the spectral clustering problem. While this results in a structured $\boldsymbol{X}$, it fails to enforce an orthogonal representation capable of separating distinct features into channels. To overcome this, we orthogonalize $\boldsymbol{X}$ by finding the eigenvectors $\boldsymbol{U}$ of a small matrix $\boldsymbol{X}^\top\boldsymbol{X} \in \mathbb{R}^{C\times C}$. This is similar to the reorthogonalization step in approximate eigensolvers; *e.g.,* lines 36-38 of Algorithm 2 in Maire and Yu [42]. The final representation is given by:

$$\boldsymbol{X}_{\text{ortho}} = \boldsymbol{X}\boldsymbol{U}. \quad (8)$$

After extracting these final features, we create hard assignments using K-Means clustering.

4. Experiments

Leveraging our method, we investigate how models group image regions internally. In Section 4.1 we see how models associate locations within an image. Section 4.2 examines the same behavior across images and discovers a spatial/semantic split depending on the choice of internal features used for grouping. We evaluate this phenomenon quantitatively, deriving a high quality training-free unsupervised semantic segmentation from Stable Diffusion [53] in Section 4.2.2, as well as providing stronger evidence for spatial information pathways in Section 4.2.3.

4.1. Per-Image Region Extraction

To show how models partition images spatially, we extract dense eigenvector for individual images and cluster these features into hard segmentations, as detailed in Section 3.2.

Experimental Setup. For all models, during optimization we consider all heads of all self-attention layers to be independent graphs. In the case of Stable Diffusion,

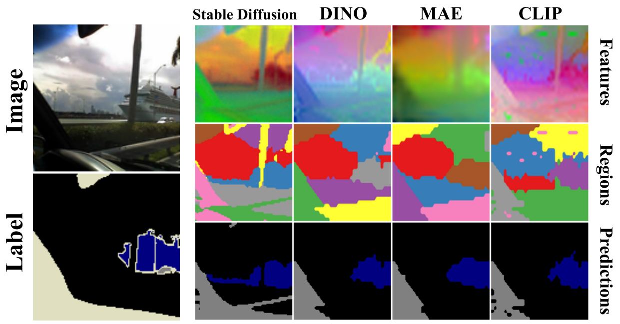


Figure 3. **Features extracted from different models on PASCAL VOC [21].** Across models we extract meaningful regions, even for models like Stable Diffusion [53], CLIP [51] or MAE [27] whose training is not well-aligned with segmentation.

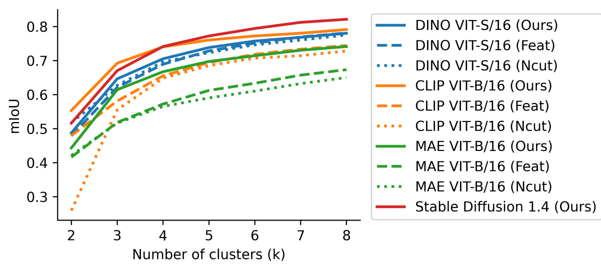


Figure 4. **Oracle-based semantic segmentation performance with varying region count.** Across models and number of clusters (regions) returned by K-Means, our method (Ours + K-Means) yields better agreement (in mIoU) with ground-truth than running Normalized Cuts (Ncut + K-Means), or directly applying K-Means on the final output features of the model (K-Means). We observe an even more significant improvement when applying our method to MAE and CLIP, which do not produce discriminative features.

this is 16 self-attention layers with 8 attention heads, thus $|\mathcal{A}| = 16 \times 8 = 128$ per forward pass. Specific to Stable Diffusion, in each iteration we add noise to the input image by randomly sampling noise timestep $t \in \mathcal{U}[0, 500)$. For all models, we construct feature map $\boldsymbol{X}$ with spatial resolution matching the finest attention layer resolution and set $C = 10$. We initialize $\boldsymbol{X}$ from a normal distribution and solve the optimization problem with Adam for ~ 2000 iterations with learning rate 1e-3.

To produce discrete regions, we run K-Means clustering by sweeping K from 2 to 10 and use silhouette score [55] to select the best value. To speed up extraction in Stable Diffusion, we cache attention matrices into a buffer for reuse with a 90% chance, bringing the per-image runtime from 154 to 67 seconds. For more implementation details, please refer to Appendix B. To provide a measure for comparison, we extract multiple regions according to two related methods: Normalized Cut [59] and MaskCut [70]. Both of these

Model	Affinity Source	Mask	mIoU
Stable Diff. 1.4 [53]	All Attentions	Ours + K-Means	**0.82**
CLIP ViT-B/16 [51]	All Attentions	Ours + K-Means	**0.78**
CLIP ViT-B/16 [51]	Final Features	K-Means	0.57
CLIP ViT-B/16 [51]	Final Features	Ncut + K-Means [71]	0.45
DINO ViT-S/16 [8]	All Attentions	Ours + K-Means	**0.78**
DINO ViT-S/16 [8]	Final Attentions	Ncut + K-Means [71]	0.58
DINO ViT-S/16 [8]	Final Features	K-Means	0.74
DINO ViT-S/16 [8]	Final Features	Ncut + K-Means [71]	0.73
DINO ViT-S/16 [8]	Final Features	MaskCut[70]	0.64
MAE ViT-B/16 [27]	All Attention	Ours + K-Means	**0.74**
MAE ViT-B/16 [27]	Final Features	Ncut + K-Means [8]	0.62
MAE ViT-B/16 [27]	Final Features	K-Means	0.48

Table 1. **Oracle decoding on PASCAL VOC [21] .** Compared with several strong baselines [70, 71] applied to single-level features, our method can consistently extract accurate segmentation. Our method works well even for models like CLIP [51] and MAE [27], whose final layer features are not discriminative enough for segmentation. Our method is agnostic to the location of information, so we avoid this difficulty.

Config	**All**	Enc	Mid	Dec	32x32	64x64
Layer Index	1-16	1-4	5 - 10	11-16	3-4 11-13	1-2 14-16
mIoU	0.75	0.66	0.76	**0.80**	0.75	0.70

$t_{\max}$	250	**500**	750	999
mIoU	**0.77**	0.75	0.74	0.69

Table 2. Ablation of layer index and maximum noise level of the diffusion model on the PASCAL VOC dataset [21]. We find that using only decoder layers and middle noise yields the best results.

methods require a single affinity matrix, the choice of which we ablate in Appendix B.

In Figure 3, we show that features and regions extracted from different models are quite structured, aligning well with object boundaries. We quantify region quality by measuring their oracle overlap with semantic segmentation labels. This gives a sense as to how well attention layers inside models decompose images along semantic axes. We perform this analysis on PASCAL VOC [21], which has 20 foreground classes and 1449 validation images. We score results with the metric of mean intersection over union (mIoU) between regions and labels. Each region is assigned to the ground-truth label it overlaps with the most.

Results and Analysis. Table 1 presents results demonstrating that our approach consistently outperforms all methods to which we compare, across various backbone models. For DINO, we show that directly clustering the final layer features using K-Means yields decent performance. This is likely due to the discriminative nature of DINO's final representation, which makes a straightforward decoding strategy sufficient for generating satisfactory regions. However, direct clustering fares much worse on other models with different training objectives.

Figure 5. **Extracted eigenvectors on COCO for both graph choices.** We visualize selected components of $\boldsymbol{X}_{\text{ortho}}$, sorted by decreasing eigenvalue. Three eigenvectors at a time are rendered as RGB images. In the Q-K case, the first set of eigenvectors describes general scene spatial layout in terms of ground, subject, background, and sky. The second finds top-to-bottom part separation within objects. In the V-V case, the first set of eigenvectors partitions the image into coarse semantics like trees, ground, and sky, while the second set recognizes finer-grained categories and groups individual objects like people, animals, and vehicles.

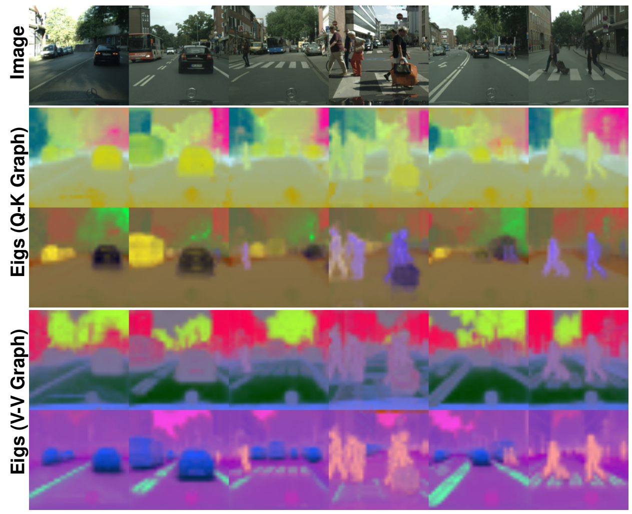


Figure 6. **Extracted eigenvectors on Cityscapes for both graph choices.** We visualize selected components of $\boldsymbol{X}_{\text{ortho}}$, sorted by decreasing eigenvalue. Three eigenvectors at a time are rendered as RGB images. In the Q-K case, eigenvectors detect the scene spatial layout and indicate how far left or right buildings, cars, trees, and people are. In the V-V case, eigenvectors perform semantic recognition and separate trees and buildings from road, and distinguish cars, people, and road markings.

Additionally, we observe that Normalized Cut [59] (Ncut) is highly sensitive to the underlying graph, and its performance deteriorates significantly when switching from the graph of final features to the final attention matrix. A related approach, MaskCut [70], solves Ncut on a binarized graph to extract foreground objects. However, this operation results in the loss of finer-grained information, which is crucial for segmentation tasks. In contrast, our method is less sensitive to the quality of a single graph because we simultaneously perform spectral clustering over a set of affinity matrices. When comparing our method on models that are not trained to produce discriminative features as their final output, such as MAE and CLIP, we observe an even more substantial improvement.

In Table 2, we provide ablation studies on the choices of layer for feature extraction and maximum noise level. Stable diffusion has 16 attention layers with resolutions from 64x64 to 8x8. Our default is All (1-16), $t_{\text{max}} = 500$. We conduct experiments on a per-image region extraction setting with 200 images from the PASCAL VOC validation set. Although our main experiment utilizes features from all layers, making minimal assumptions about layer-wise feature distribution, we find that using only decoder layers and a middle noise level yields better results.

To further evaluate the region quality irrespective of decoding choices, Figure 4 shows the mIoU with varying choice of K. We see that the high quality of regions persists across choices, even when compared with baselines.

Our per-image regions can find broad applicability in a variety of segmentation tasks. For first-step proof-of-concepts, see Appendices D.1 and D.2.

4.2. Full-Dataset Region Extraction

Our method can effectively extract regions within images. Can it examine relationships across images? To probe different kinds of encoded information, we take the best model of the previous section, Stable Diffusion, as a case-study. We compare the query/key (Q-K) dataset-level graph with the value/value (V-V) dataset-level graph, as described in Section 3.3. Results show a surprisingly structured split, where Q-K encodes spatial information and V-V encodes semantic information, which we can use for tasks like unsupervised semantic segmentation.

Experimental Setup. For constructing graphs, we follow the method in Section 3.3. For efficiency, we concatenate features at each head into a single vector instead of considering heads independently. We select one attention block in the middle block and the first 6 attention blocks in the upsampling blocks, resulting in a total of 7 attention matrices. We choose channel number $C = 50$, cross-image connections $c_{\text{inter}} = c_{\text{intra}} = 10$, noise level $t \in \mathcal{U}[20, 300)$, and optimize using Adam [33] with a learning rate of 1e-2, and a batch size of 160 images over 4 GPUs for 2100 iterations.

When clustering, we choose K to be the number of labels of the relevant task. More details are in Appendix C.

4.2.1 Qualitative Analysis

We show qualitative results for both Q-K and V-V graphs on COCO [40] in Figure 5 and Cityscapes [16] in Figure 6.

Across datasets, we observe that the Q-K graph appears to encode spatial relationships. On Cityscapes, which has a clear spatial layout at the scene level, the learned eigenvectors effectively separate buildings, cars, people, and trees into left/right subgroups. For the more complex dataset COCO, which lacks fixed spatial patterns at the scene level, the eigenvectors uncover spatial correlations first in terms of ground, subject, and background, and then part-like correlations within objects from top-to-bottom.

By contrast, features from the V-V graph group objects semantically. In COCO, we observe that eigenvectors encode semantic structure hierarchically: the first set of eigenvectors focuses on distinguishing scene-level semantics (*e.g.,* ground, sky, trees) while overlooking differentiating foreground objects. The next set of eigenvectors groups foreground objects like people, animals, and vehicles. In Cityscapes, the initial set captures broad scene-level semantics, including trees, houses, and the egocentric vehicle, and can differentiate between road and sidewalk. The following set groups cars, people, and road markings. More examples are available in Appendix A.

The qualitative differences between eigenvectors stemming from the Q-K and V-V graphs suggest a clear split in the way the model processes information from images. In attention layers, queries and keys are used to form patch-wise relationships, which then modulate the values propagated to the next stage. It appears that the model learns to split representation into spatial and semantic branches as a convenient solution for taking advantage of this computational structure. This is perhaps surprising, as the projection from features to queries or keys or values need not split information in such a clean fashion. The discrepancy in the behavior between features from these different graphs motivates us to further evaluate their performance in separate semantic and spatial benchmarks.

4.2.2 Quantitative Analysis for 'What' Pathway

To quantify semantic segmentation ability in the V-V graph, we evaluate our extracted segmentation on two common unsupervised semantic segmentation tasks: COCO-Stuff [5, 40] and Cityscapes [16]. We follow the preprocessing protocol as adopted in PiCIE [14] and STEGO [24]. We optimize $\boldsymbol{X}$ on the validation set, where images are first resized so the minor edge is 320px and then cropped in the center to produce square images. We choose $K = 27$, the number of ground-truth categories in both datasets, for K-Means over $\boldsymbol{X}$, and then use greedy matching to align the cluster assignments with the ground truth. We report results with mIoU and compare to other methods in Table 3, and examine feature choice and decoding protocol in Table 4. More details are in Appendix C.

Method	**Results (mIOU)**	
	COCO-Stuff-27	**Cityscapes**
MoCo v2 [12]	4.4	-
IIC [30]	6.7	6.1
DSM [47] (ViT-B/8)	8.9	-
Modified DC [14]	9.8	7.4
PiCIE [14]	13.8	12.3
PiCIE+H [14]	14.4	-
ACSeg [39]	16.4	-
HP [58] (ViT-S/8)	24.6	18.4
STEGO [24] (ViT-B/8)	26.8	18.2
STEGO [24]+CRF (ViT-B/8)	**28.2**	**21.0**
Ours (V-V graph)	25.4	16.2
Ours (V-V graph)+CRF	27.1	16.9

Table 3. **Unsupervised semantic segmentation results on COCO-Stuff-27 and Cityscapes.** We observe that the V-V graph features outperform those of prior works and achieve competitive performance compared to the strong STEGO method, which utilizes discriminative DINO [8] features and a complex two-stage global nearest-neighbor strategy. Conversely, our method employs representations from a generative model and collects neighbors solely from the minibatch, a simpler and more scalable approach.

In Table 3, our method significantly outperforms many other methods and is comparable to STEGO [24]. This is quite surprising, as STEGO [24] adopts a sophisticated two-stage dataset-wise nearest-neighbor searching procedure, while our method only considers connections within the mini-batch, a strategy with noisy signal but with better scalability. STEGO [24] also benefits from the discriminative representations of DINO [8], while our backbone, Stable Diffusion [53], is generative.

Table 4 reports results on directly clustering the most semantic representations of Stable Diffusion [53], which are the features of the 2nd upsampling block with timestep $t = 250$ [63]. In this comparison, DINO [8] features are better than Stable Diffusion, likely due to their discriminative properties, but our method greatly narrows the gap.

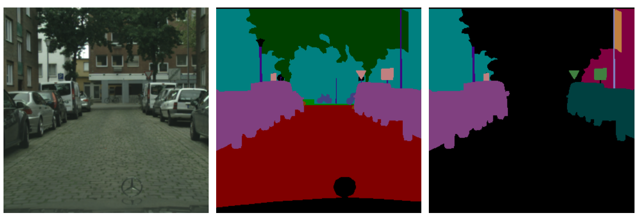
4.2.3 Quantitative Analysis for 'Where' Pathway

Here, we design two experiments to quantify the positional information in the Q-K graph for the 'where' pathway. Our first experiment aims to measure the amount of positional information contained within the features. In this setting, we train a linear head on top of the features and attempt to regress the corresponding grid position of each pixel/patch. We call this task "coordinate regression".

The second experiment aims to evaluate whether this spatial information is present at the semantic level. In this

Method	COCO-Stuff-27		Cityscapes	
	Greedy	Hungarian	Greedy	Hungarian
K-Means (SD [53])	9.2	8.6	12.4	8.1
K-Means (DINO [8])	13.7	13.0	13.3	8.7
K-Means (STEGO [24])	26.6	24.0	15.8	14.9
STEGO [24]	**27.0**	**26.5**	**16.6**	**18.2**
Ours (Q-K graph)	12.4	10.9	10.91	9.7
Ours (V-V graph)	25.4	23.2	16.2	11.4

Table 4. **Ablations for unsupervised semantic segmentation.** We test multiple sources of features for clustering on the validation set only, and vary the decoding pipeline for evaluation. With greedy decoding, our features are comparable, but with Hungarian matching STEGO is stronger. We also see that the Q-K graph encodes far less semantic information than the V-V graph, supporting a semantic/spatial decomposition. The discrepancy between K-Means (STEGO) and STEGO numbers here is due to restricting clustering to the validation set.



(a) Image (b) Semantic Label (c) Processed Label

Figure 7. **Spatial semantic segmentation task.** We generate labels for "spatial semantic segmentation" by splitting the semantic labels into left/right subgroups, followed by filtering out small regions and ambiguous regions close to the image center.

case, we benchmark on an unsupervised semantic segmentation task by further partitioning the semantic annotations into left and right subgroups. For this purpose, we process the ground-truth annotations by first identifying disconnected regions for each category of segmentation annotation and then scoring each region based on its pixel distance to the image's left/right border. We filter out smaller regions with pixel counts less than 50 and ambiguous regions located close to the image center. We call this task "spatial semantic segmentation" and showcase the original and processed semantic segmentation maps in Figure 7. We follow the exact same evaluation protocol as used in the experiment for the 'what' pathway.

The results of both experiments are presented in Table 5, where we compare with STEGO features, DINO final-layer features, and ground-truth semantic segmentation labels. In both experiments, our approach with the Q-K graph outperforms both STEGO and the variants with the V-V graph.

Results on coordinate regression suggest that $\boldsymbol{X}$ from the Q-K graph contains rich spatial information for processing the 'where' pathway. However, both STEGO and the V-V graph group pixels only by semantic similarity and remove spatial information from the final representation. DINO performs well on regression likely due to position embeddings.

Method	Coordinate Regression (MSE) ↓	Spatial Semantic Segmentation (mIOU) ↑
DINO	**3.2**	6.0
GT semantic label	72.0	-
STEGO	42.4	6.9
Ours (V-V graph)	43.1	5.1
Ours (Q-K graph)	**19.5**	**10.1**

Table 5. **Results for evaluating 'where' pathway on spatial structures**. Ours (Q-K graph) outperforms STEGO and ours (V-V graph) in both benchmarks suggesting the Q-K graph contains richer spatial information at object levels. Though DINO can trivially recover the spatial coordinates through positional embeddings, it fails to leverage that information for segmentation.

We further verify the spatial information content of the Q-K graph by examining the results of spatial semantic segmentation. We see that these features are strongest in this task. Compared to results on unsupervised semantic segmentation (Table 3), the strong performance of features from the V-V graph and STEGO deteriorates due to failure to reason about spatial structure. DINO features also fail in this task, likely as spatial information is not as strong a signal as semantics for discriminating between images. These results, along with those in Section 4.2.2, show that our approach can scale efficiently to extract both spatial and semantic relationships across images.

5. Discussion

We present an approach for extracting information from a neural network's activations. Unlike prior work, our method examines the whole of a network, without needing to guess which part of the model contains relevant features. Our approach resembles classic spectral clustering, but gains scalability to dataset-level analysis by approximating a solution using gradient-based optimization.

Deployed as a mechanism for extracting image segmentation from large pre-trained models, we observe robust performance in producing regions from a wide variety of source models, including high quality semantic segmentations obtained from a Stable Diffusion model. Deployed as an analysis tool, we gain new insight into how vision models with attention layers utilize key, query, and value vectors to coordinate the flow of spatial and semantic information, and disentangle 'what' and 'where' pathways within these deep networks.

Our approach could be the first example in a new class of optimization-centric techniques for peering into the inner workings of deep networks. Future research could repurpose other computationally intensive, but scalable, classic machine learning tools to the analysis of network behavior.

Acknowledgments. NSF GRFP (1754881) supported D. Yunis.

References

[1] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011.

[2] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2021.

[3] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. In *ICCV*, 2015.

[4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021.

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018.

[6] John Canny. A computational approach to edge detection. *PAMI*, 1986.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[9] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *ICCV*, 2021.

[10] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *CVPR*, 2021.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020.

[13] Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv:2306.05720*, 2023.

[14] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. PiCIE: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *CVPR*, 2021.

[15] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT's attention. *arXiv:1906.04341*, 2019.

[16] 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 *CVPR*, 2016.

[17] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.

[18] Piotr Dollár and C. Lawrence Zitnick. Fast edge detection using structured forests. *PAMI*, 2015.

[19] Mischa Dombrowski, Hadrien Reynaud, Matthew Baugh, and Bernhard Kainz. Foreground-background separation through concept distillation from generative image foundation models. In *ICCV*, 2023.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[21] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 2015.

[22] Melvyn A. Goodale and A. David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 1992.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[24] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022.

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[28] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *arXiv:2305.15581*, 2023.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[30] Xu Ji, Joao F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

[31] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up GANs for text-to-image synthesis. In *CVPR*, 2023.

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

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

[34] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

[35] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv:1511.07386*, 2015.

[36] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. In *EMNLP*, 2019.

[37] Dwight J. Kravitz, Kadharbatcha S. Saleem, Chris I. Baker, and Mortimer Mishkin. A new neural framework for visuospatial processing. *Nature Reviews Neuroscience*, 2011.

[38] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022.

[39] Jiahao Li, Greg Shakhnarovich, and Raymond A. Yeh. Adapting CLIP for phrase localization without further training. *arXiv:2204.03647*, 2022.

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[41] Krzysztof Lis, Matthias Rottmann, Sina Honari, Pascal Fua, and Mathieu Salzmann. AttEntropy: Segmenting unknown objects in complex scenes using the spatial attention entropy of semantic segmentation transformers. *arXiv:2212.14397*, 2022.

[42] Michael Maire and Stella X Yu. Progressive multigrid eigensolvers for multiscale spectral segmentation. In *ICCV*, 2013.

[43] Michael Maire, Stella X. Yu, and Pietro Perona. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011.

[44] Michael Maire, Takuya Narihira, and Stella X. Yu. Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding. In *CVPR*, 2016.

[45] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[46] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *NeurIPS*, 2000.

[47] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022.

[48] Mortimer Mishkin, Leslie G. Ungerleider, and Kathleen A. Macko. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 1983.

[49] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: An overview. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 2019.

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[51] 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 *ICML*, 2021.

[52] Xiaofeng Ren and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. In *NeurIPS*, 2012.

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[55] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987.

[56] Gerald E. Schneider. Two visual systems: Brain mechanisms for localization and discrimination are dissociated by tectal and cortical lesions. *Science*, 1969.

[57] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[58] Hyun Seok Seong, WonJun Moon, SuBeen Lee, and Jae-Pil Heo. Leveraging hidden positives for unsupervised semantic segmentation. In *CVPR*, 2023.

[59] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[60] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[61] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv:1412.6806*, 2014.

[62] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.

[63] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *NeurIPS*, 2023.

[64] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised CNN segmentation. In *CVPR*, 2018.

[65] Colwyn B. Trevarthen. Two mechanisms of vision in primates. *Psychologische Forschung*, 1968.

[66] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[68] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[69] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M. Alvarez. FreeSOLO: Learning to segment objects without annotations. In *CVPR*, 2022.

[70] Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *CVPR*, 2023.

[71] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L. Crowley, and Dominique Vaufreydaz. TokenCut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *arXiv:2209.00383*, 2022.

[72] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[73] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic segmentation emerges from text supervision. In *CVPR*, 2022.

[74] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[75] Stella X. Yu and Jianbo Shi. Segmentation given partial grouping constraints. *PAMI*, 2004.

[76] Stella X. Yu, Ralph Gross, and Jianbo Shi. Concurrent object recognition and segmentation by graph partitioning. In *NeurIPS*, 2002.

[77] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *ECCV*, 2022.

[78] Adrian Ziegler and Yuki M. Asano. Self-supervised learning of object parts for semantic segmentation. In *CVPR*, 2022.