Learning Visual Groupings and Representations with Minimal Human Labels

by

Tsung-Wei Ke

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Vision Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Stella X. Yu, Co-chair
Professor David Whitney, Co-chair
Professor Alexei A. Efros
Professor Meng Lin

Fall 2022

Learning Visual Groupings and Representations with Minimal Human Labels

Abstract

Learning Visual Groupings and Representations with Minimal Human Labels

by

Tsung-Wei Ke

Doctor of Philosophy in Vision Science

University of California, Berkeley

Professor Stella X. Yu, Co-chair

Professor David Whitney, Co-chair

Making a computer system understand complex image scenes is challenging. Complex image scenes often have multiple objects, which are not isolated but related to each other in different aspects. Identifying certain object categories may not be enough to understand complex scenes. Categories have multiple granularities. We need such knowledge to capture semantic correlation thoroughly. In addition, objects have numerous interactions/relationships. We need to localize these objects, recognize scene environments, and figure out their interactions/relationships. In computer vision, recognizing *what the categories are*, *where the objects are*, and *how objects interact to each other* is often formulated as the classification, segmentation, and relationship recognition problem.

Existing approaches often tackle all these formulations in supervised settings. Despite their tremendous progress, we identify three major limitations. **1)** Human annotation is too time- and labor-consuming to scale up to real-world scenarios. **2)** The sets of human labels are pre-selected arbitrarily, providing limited/biased perspectives to understand images. **3)** Such supervised methods conduct inference in terms of discrete labeling. They isolate labels from each other, ignoring the similarity/dissimilarity among each other. Also, they can only put images to the known labels seen during training and fail to recognize novel images sampled from unknown labels during testing.

In this dissertation, we address the issues of current supervised approaches by replacing discrete labeling with grouping and using minimal human labels. Specifically, we tackle the recognition problem from four perspectives. **1)** We address weakly-supervised semantic segmentation, where partial semantic pixel labels are used. **2)** We address unsupervised semantic segmentation, where only low-level edge detections are used. **3)** We address unsupervised concurrent image classification and segmentation in a single framework, where our model does not use any human labels. **4)** We address unsupervised human-object

recognition, where semantic and instance pixels labels, no relationship labels, are used. This dissertation explores more general and robust approaches to understanding the highly-complex and fast-changing real-world scene.

To my family for their love and support

Contents

List of Figures

List of Tables

Acknowledgments

I would not be able to complete my Ph.D. study and this dissertation without the support and guidance of my advisors, Professor Stella Yu and Professor David Whitney.

I would like to thank Stella for her guidance and support throughout my years at Berkeley. Six years ago, when I struggled to switch my career from chemical engineering to computer vision, Stella offered me an opportunity and we began this journey at ICSI, Berkeley. She becomes my role model of great researchers and thinkers. She tells me that research is not merely solving individual problems, more importantly, finding the connections and distilling the core ideas among them. She says papers are not for reporting current techniques but for elaborating these core ideas that will inspire people in the future. I hope that I have met the minimal standard in these aspects.

I would like to thank David for his inspiration from the side of human vision. Before joining Berkeley, I was arrogant to believe that computers would soon surpass humans and did not appreciate human vision research. It turns out I was so wrong. For our interdisciplinary projects, my lab rotation, qualification exam, and this dissertation, David not only helps me to go through human vision literature but also provides interesting ideas from human-vision perspectives. All these knowledge and ideas have inspired my exploration and formulation of unsupervised recognition problems in this dissertation. I would not complete this dissertation without David's insights into human vision.

I would also like to thank my mentors and collaborators whom I was fortunate to work with throughout this journey. Foremost, I am super lucky to have Jyh-Jing as my academic brother and close friend, from whom I learn how to think, speak, and write as a professional researcher. I appreciate his guidance and mentorship in both aspects of research and life. Alex and Henrik at Waymo provided an opportunity for me to get a clear sense of industrial applications. Liang and Ren at Bosch were supportive of bridging the gap between industrial and academic research. Finally, I really enjoy working together with Sungbum/Soon-Young from Samsung, Nicholas/Chao/Daniela from LBL, and Mark/Jennifer/Enrika/Kyle/Luke from USGS on different application projects.

I would like to thank my friends at Berkeley and my labmates in Stella's group. I enjoyed hanging out with Kuan-Yun, Zhongqi, and Jyh-Jing over the weekends and holidays, which really helped me to overcome loneliness when studying abroad. I miss having lunch with Pat, Sascha, Mario, Matthias, Ziwei, Zhirong, Qian, Bala, Peters, Xudong, Yunhui, Utkarsh, and Nils at ICSI before the pandemic. It was a good time.

Lastly, I would like to devote my gratefulness to my family back in Taiwan. Without their physical, financial, and emotional support, I would never be able to get through low points and trounce frustrations throughout this journey. I had been frustrated and doubting myself. But every time traveling back from Taiwan, I am always refreshed and ready to confront the next challenges. I am lucky to have them get my back.

Chapter 1

Introduction

Making a computer system understand complex image scenes is challenging. When the scene has only one centered object (the left in Fig. 1.1), the task is as simple as identifying the object's category, e.g. a meerkat. When the scene is composed of multiple objects (the right in Fig. 1.1), knowing certain object categories may not be enough to understand the image. *What do the woman and dogs wear?* To answer the question, we need not only recognize *clothes* category but also have the knowledge that categories have different granularities (both *swimsuit* and *shirt* are *clothes*). *What are the woman and dogs doing?* To answer the question, we need to localize the objects, identify scene environments, and figure out their relationships/interactions.

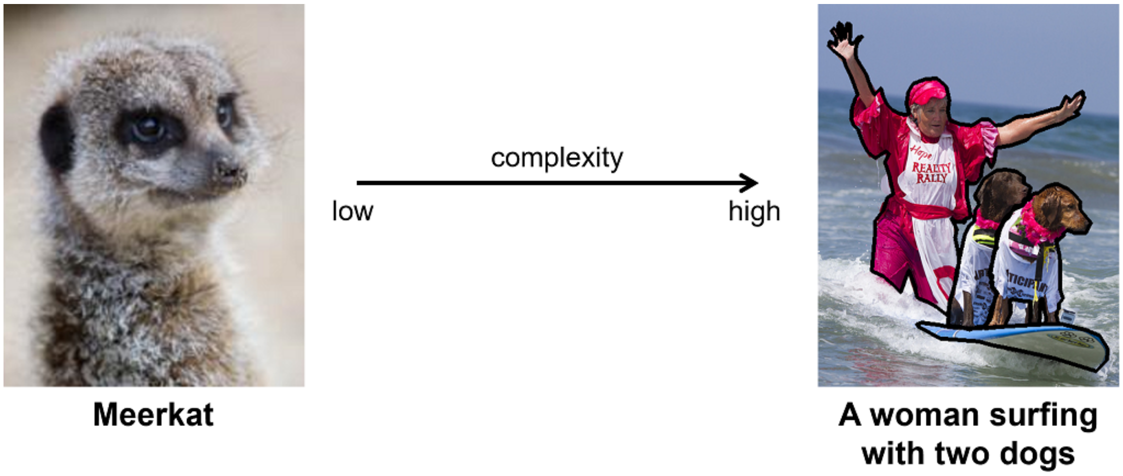


Figure 1.1: Making a computer system understand complex image scenes is challenging. **Left:** A scene has a centered object, where recognition is as simple as naming the object's category. **Right:** A scene has multiple objects, where recognition is more difficult. *What do the woman and dogs wear? What are the woman and dogs doing?* Naming certain objects' categories is not enough to answer these questions.

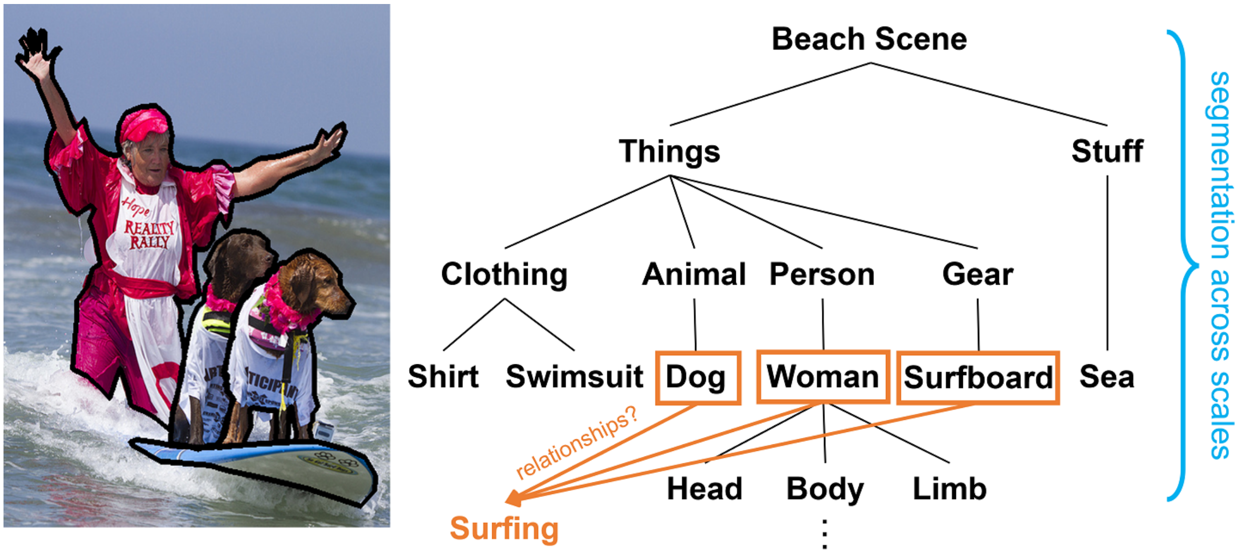


Figure 1.2: Understanding complex scenes involves different aspects of recognition. To answer *what do the woman and dogs wear?*, we need not only recognize the *clothes* category but also know that categories have different granularities (*swimsuit* and *shirts* are *clothes*). To answer *what are the woman and dogs doings?*, we need to localize objects and figure out their relationships. In particular, segmentation provides more fine-grained and general localization that locates both objects and non-object components at the pixel level. We segment an image into different regions, where each region includes pixels belonging to the same category/instance. Segmentations may have different granularities to enable recognition at different scales. We can thus pick out the body parts from the woman, distinguish the woman from two dogs, and separate foreground objects from the sea scene.

In computer vision, recognizing what's in the image has often been formulated as image classification, where deep learning techniques [1, 2, 3, 4] have made tremendous progress. There have been several lines of approaches: **1)** template-matching methods [5, 6] create image templates for each category and classify inputs by matching with these templates, **2)** feature discrimination methods [7, 8] extract hand-crafted visual features and apply discriminative models to classify images, and **3)** part-based constellation methods [9, 10] model the spatial configuration of each category and classify by matching the spatial layout. For example, a tree consists of leaves (top) and a trunk (bottom). Recently, with the availability of large-scale labeled imagery, e.g. ImageNet [11] has 1.3 million images labeled in 1000 pre-defined classes. Deep learning techniques have shown their superior performance over other methods. In fact, ViT [4] achieves less than 2% classification error compared to 5% from human experts [12] on ImageNet.

However, such classification methods only predict *what*, not *where*, the object categories are in images. A common object localization approach is to draw the boxes that tightly enclose each instance [13, 14]. Yet, such box-based predictions provide only coarse location information and ignore non-object components in the image, e.g. sea and sky.

A more fine-grained and general localization approach is segmentation, where we locate both objects and non-object components at the pixel level. The image segmentation task is to separate all image pixels into fewer groups (regions), where pixels belonging to the same category and instance are put into the same group [15, 16, 17]. We can thus pick out the woman, dogs, the sky, and the sea pixels from the image. Notably, segmentations are not restricted to the pre-defined semantics but are applicable to humans' general sense of visual perception [18]. In addition, we can perform segmentation at different levels of granularity to enable recognition at different scales. We can thus pick out the body parts from the woman, distinguish the woman from two dogs, and separate foreground objects from the sea scene.

After isolating objects in the scene, the next step is to figure out their relationships [19, 20]. In particular, we need not consider individual objects in isolation but joint configurations of objects and surroundings, a.k.a visual context. Visual context has been modeled differently, capturing co-occurring semantics [21, 22], statistics [23], spatial layout [24], *etc.* With such information, we can discriminate the person riding a horse from the person driving a car. Recently, relationship recognition is formulated as a pair-wise classification problem [25, 26]: a model enumerates all combinations of human and object pairs and predicts their relationship category, respectively.

Existing methods often tackle all these formulations: classification, segmentation, and

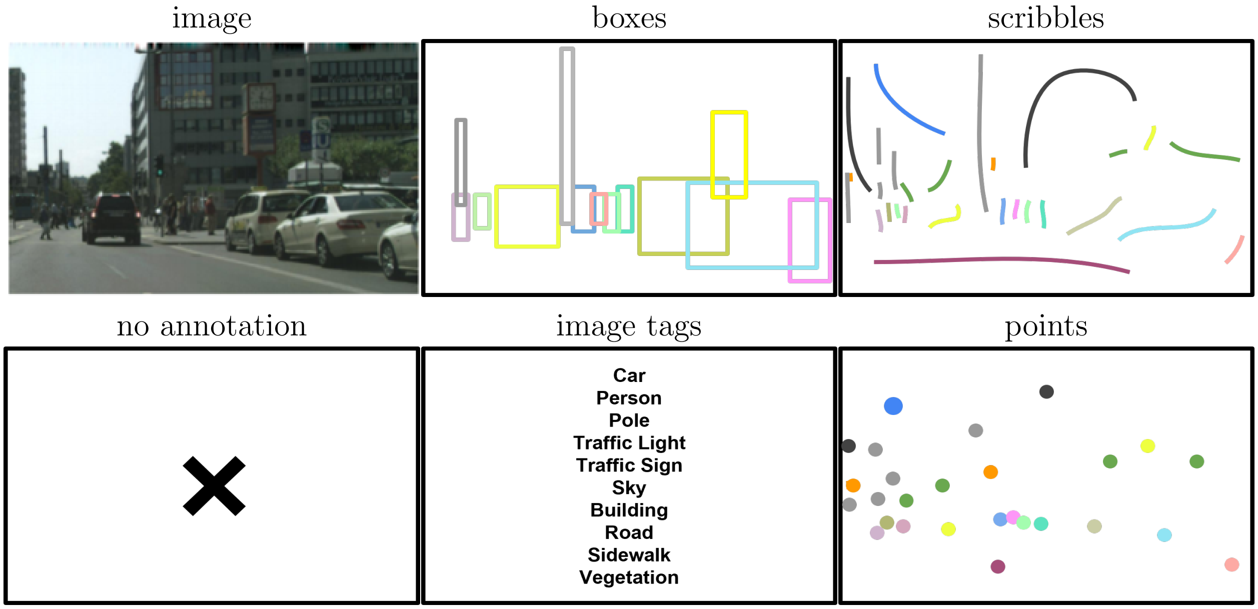


Figure 1.3: Different forms of simple but imprecise annotation carry different assumptions. Image tags and bounding boxes lack precise localization information, while scribbles and points are sparsely annotated. Each type of annotation requires different techniques for training. We instead present a single framework to handle all types of partial labels jointly.

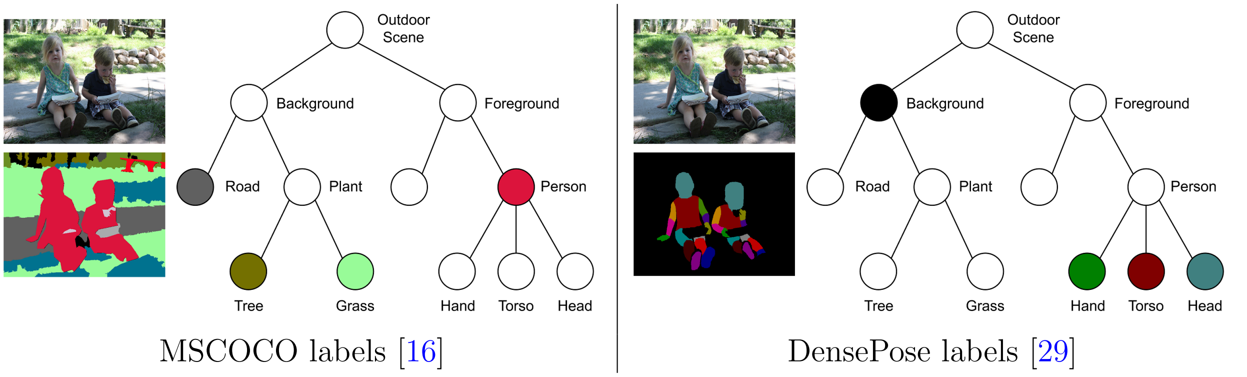


Figure 1.4: The sets of human labels are pre-selected arbitrarily, ignoring the ambiguity of granularity and providing limited perspectives to understand images. **Left:** Semantic classes annotated on MSCOCO [16]. Such a label set has no knowledge of human body parts. **Right:** Semantic classes annotated on DensePose [16]. Such a label set oversimplifies diverse stuff (e.g. grass, tree, and road) into a background class. Learning from either set of labels results in biased and limited image recognition.

relationship recognition in supervised settings, where models are trained and tested on images sampled from a fixed set of human labels. Though such supervised approaches have achieved remarkable performance, they have three major limitations.

First, human annotation is too time- and labor-consuming to scale up to real-world scenarios, where we have enormous amounts of images in highly complex scenes. It takes hours to obtain high-quality segmentation labels on a high-resolution image [27]. Though different kinds of simple annotations are proposed to address the issue (Fig. 1.3), they are imprecise and carry different assumptions. Tags and boxes lack precise pixel localization and provide coarse supervision, whereas points and scribbles annotate only a subset of pixels and provide sparse supervision. On the other hand, annotating complete relationships in complex scenes is also impractical. The number of actual relationships grows exponentially and quickly explodes with increasing numbers of object categories and instances. For example, Visual Genome [28] already has $33,877$ different object categories and $42,374$ types of *pair-wise* object relationships. However, the dataset has not yet taken *group-wise* relationships into account, which will otherwise increase the number of relationships drastically. It is thus infeasible to build and annotate a dataset on a real-world scale for both segmentation and relationship recognition.

Secondly, the sets of human labels are pre-selected arbitrarily, ignoring the ambiguity of granularity and providing limited perspectives to understand images. Take semantic segmentation for example. As shown in Fig. 1.4, on MSCOCO [16], images are parsed into things and stuff classes (e.g. *person*, *grass*, *tree*, and *road*). Such a label set has no knowledge of finer-grained categories, e.g. human body parts. On DensePose [29], images are parsed into body-part categories and a background class, where all kinds of stuff are over-simplified

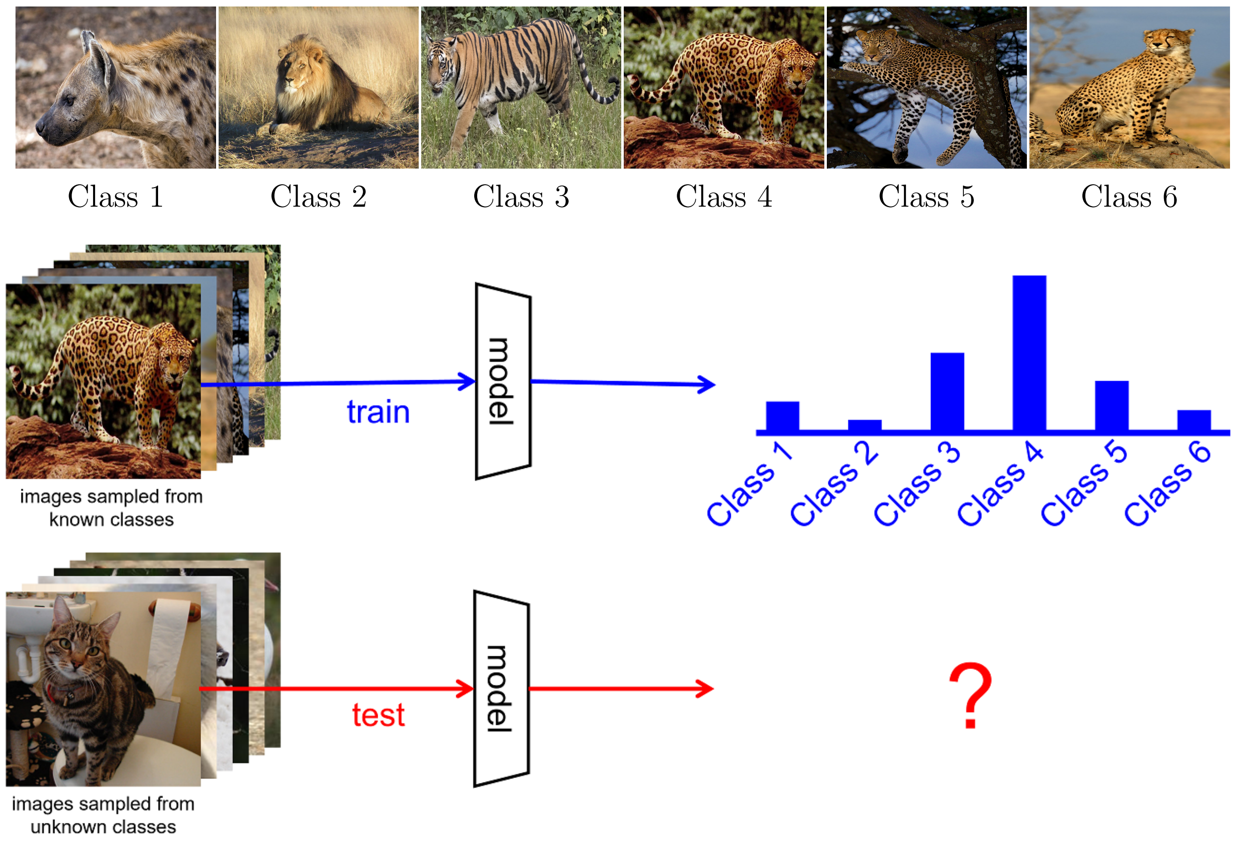


Figure 1.5: Existing supervised models conduct inference in terms of discrete labeling. Take classification, for example, the classifier puts input images into isolated classes. Yet, the model does not know similarity/dissimilarity among categories and fails to recognize images of unseen classes. **Top:** *Hyenas*, *lion*, *tiger*, *jaguar*, *leopard*, and *cheetah* are considered as 6 discrete classes, disregarding the fact that jaguars are more similar to leopards and cheetahs than hyenas. **Bottom:** Models can only classify images into one of the classes that are known during training. They fail to recognize novel images sampled from unknown labels during testing. Such a kind of supervised modeling can not explain relationships among images. It is also not applicable to real-world environments, where image scenes could come from unknown distributions.

as background. Such a label set ignores the diversity of backgrounds. Learning from either set of annotations results in biased and limited image recognition.

Lastly, existing supervised methods often conduct inference in terms of discrete labeling, which considers labels to be isolated from each other and ignores similarity and dissimilarity among them. Take supervised classification for example (Fig. 1.5), the model classifies images into 6 discrete classes without knowing correlations among *hyenas*, *lion*, *tiger*, *jaguar*, *leopard*, and *cheetah*. In addition, the learning objective is to discriminate each class equally,

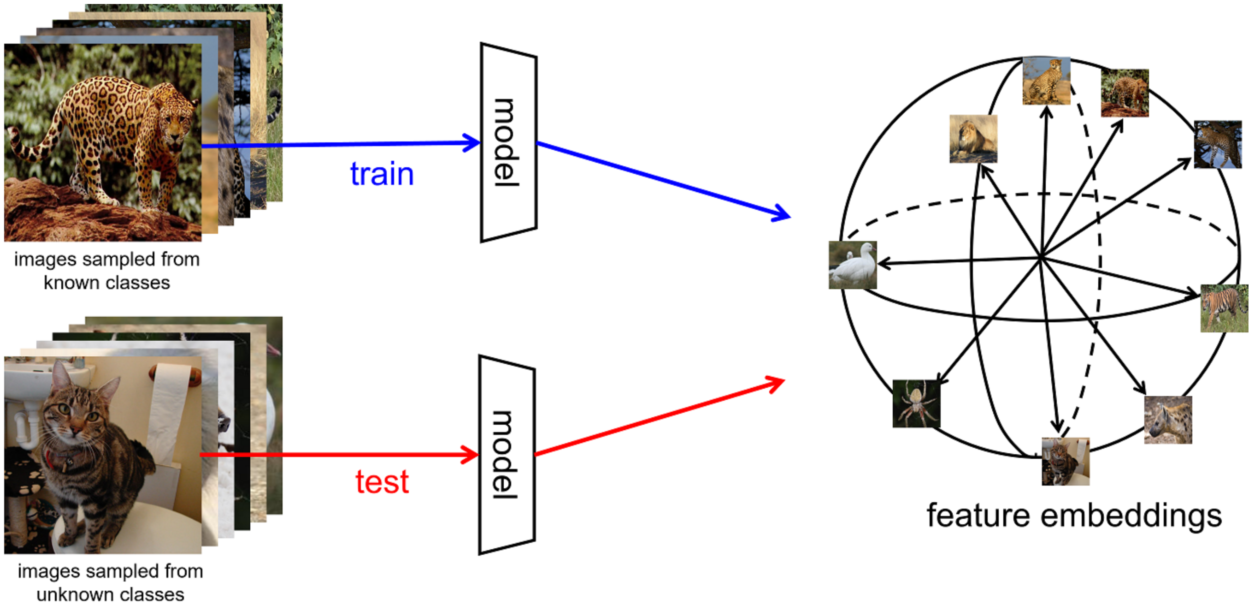

images sampled from known classes
train
model
images sampled from unknown classes
test
model
feature embeddings

Figure 1.6: Our grouping frameworks map images/pixels into continuous feature points, not discrete labels, to perform inference. Distances in the latent feature space correspond to similarities in the input image space. For classification, segmentation, and relationship recognition, images, pixels, and objects are grouped if they have the same categories, belong to the same segments, and carry the same relationships. The models output continuous feature representations which are more interpretable than discrete labels. We can infer correlations among inputs based on their feature distances. Such methods are more robust to *open-set* recognition, where images may come from unknown distributions. Novel images are projected to the same latent feature space, and we can recognize them by looking up their feature similarities with all the others.

disregarding the fact that *leopard* is more similar to *jaguar* than *hyenas*. We are thus restricted from sharing information among similar classes.

Moreover, such a kind of supervised modeling can only put images to the known labels seen during training, which fails to recognize novel images sampled from unknown labels during testing, e.g. *cat*. Their applications to real-world environments, where image scenes could come from unknown distributions, are limited.

To address these issues, we explore an alternative to the learning paradigm of using pre-defined and discrete human labels. We do not use human annotations for training. Instead, the idea of unsupervised learning is to supervise models with *a priori* or data-driven knowledge. Models are thus not biased by arbitrary relationships encoded in human labels and enforced to capture rich information in images. Importantly, models can enjoy huge amounts of data without constraints of annotation costs.

As for modeling, we transform the idea of tagging into grouping: our models do not put images/pixels into discrete labels but map them into continuous feature points (Fig. 1.6).

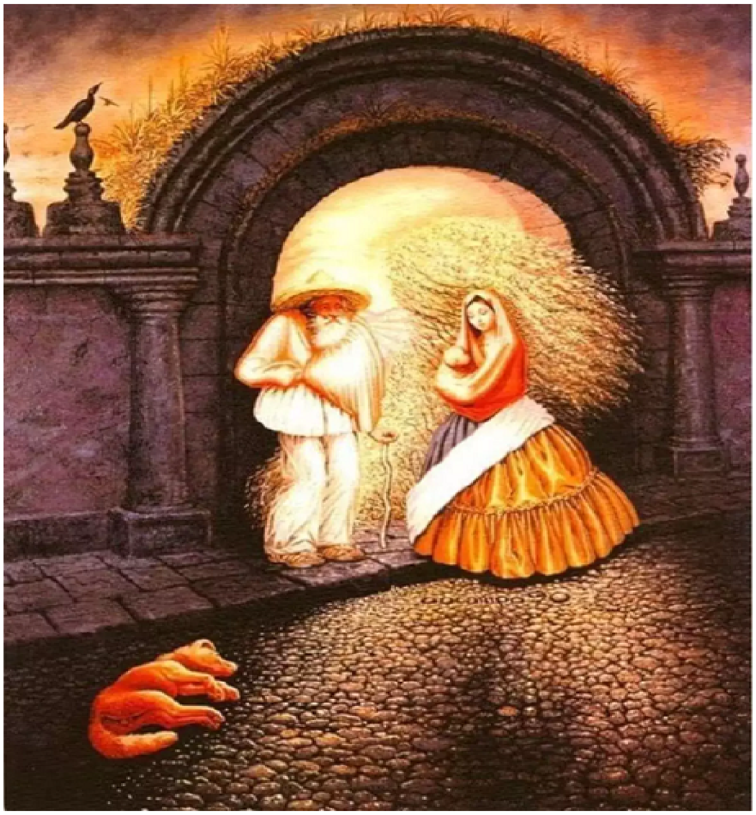


Figure 1.7: Whole-image recognition and segmentation are not separate but intertwined. Groupings of image pixels vary with our perception of the image. If we see an old general, he wears a golden epaulet and has a red ear; if we see an old man and a young lady, the general's ear and epaulet become the lady's red clothes and yellow skirts. This dissertation proposes a computational model that handles both tasks simultaneously and consistently. Image source: *The General's Family* by Octavio Ocampo.

Distances in the latent feature space correspond to similarities in the input image space. For classification, same-(different-) category images are closer (farther); for segmentation, same-(different-) component pixels are grouped (separated). Moreover, we can learn feature representations to recognize relationships/interactions among image components. Notably, such methods are more general to *open-set* recognition, where images may come from unknown labels. Without the need for knowing new labels, we project novel images onto the same feature space and recognize them by looking up their feature similarities with all the others.

We next explore a grouping framework to unify classification and segmentation. Most existing methods tackle them separately: classification/segmentation results are independent of each other. Though we can isolate these two tasks, there is still ambiguity: how you interpret/recognize the image will also affect the segmentations. Fig. 1.7 shows an example. Let's focus on the center of the image. If we see an old general, the yellow (red) region becomes his golden epaulet (red ear). In contrast, if we see a man and a woman, the yellow and red region becomes the woman's red clothes and yellow skirts. This example of illusion showcases the entanglement of classification and segmentation: pixel groupings are adaptive to our image-wise recognition. We are thus motivated to propose a unified grouping framework to conduct both tasks, concurrently.

1.1 Dissertation Overview

In this dissertation, we approach classification, segmentation, and relationship recognition problems with minimal human labels: **1)** weakly-supervised semantic segmentation [30], **2)** unsupervised hierarchical segmentation [31], **3)** unsupervised classification and segmentation [32], and **4)** unsupervised human-object relationship recognition [33]. We illustrate our research roadmap in Fig. 1.8.

Universal Weakly Supervised Segmentation

Chapter 2 presents an approach to deal with universal weakly supervised semantic segmentation. Current methods utilize different techniques to tackle each type of weak annotation (e.g. boxes, scribbles, points, and image tags) separately. Instead, we use a single contrastive loss formulation to integrate all types of weak annotations, albeit they have different assumptions. We propose the first framework that can deal with all kinds of weak annotations jointly, and when using one type of annotation at a time, our method also outperforms baselines by a large margin.

Unsupervised Hierarchical Semantic Segmentation

Chapter 3 describes a general method for unsupervised hierarchical semantic segmentation. Without human-labeled supervision, the task of unsupervised semantic segmentation is to group, not classify, pixels in unlabeled images. Groupings intrinsically have multiple scales, whereas existing methods ignore the ambiguity and perform grouping at a single level of granularity. We instead embrace the ambiguity of granularity and enforce consistency across scales to develop our feature representations. Our model produces hierarchical segmentations for input images, which capture semantics more precisely than baselines at each scale.

Unsupervised Concurrent Classification and Segmentation

Chapter 4 introduces an unsupervised grouping framework that unifies classification and segmentation. Previous works tackle these two tasks separately and connect them with staged transfer learning procedures. In contrast, we innovate Vision Transformer architectures [4] to perform both tasks concurrently. In particular, we use segment, not patch, tokens throughout the model. We create a token hierarchy by merging segment tokens into larger-region tokens,

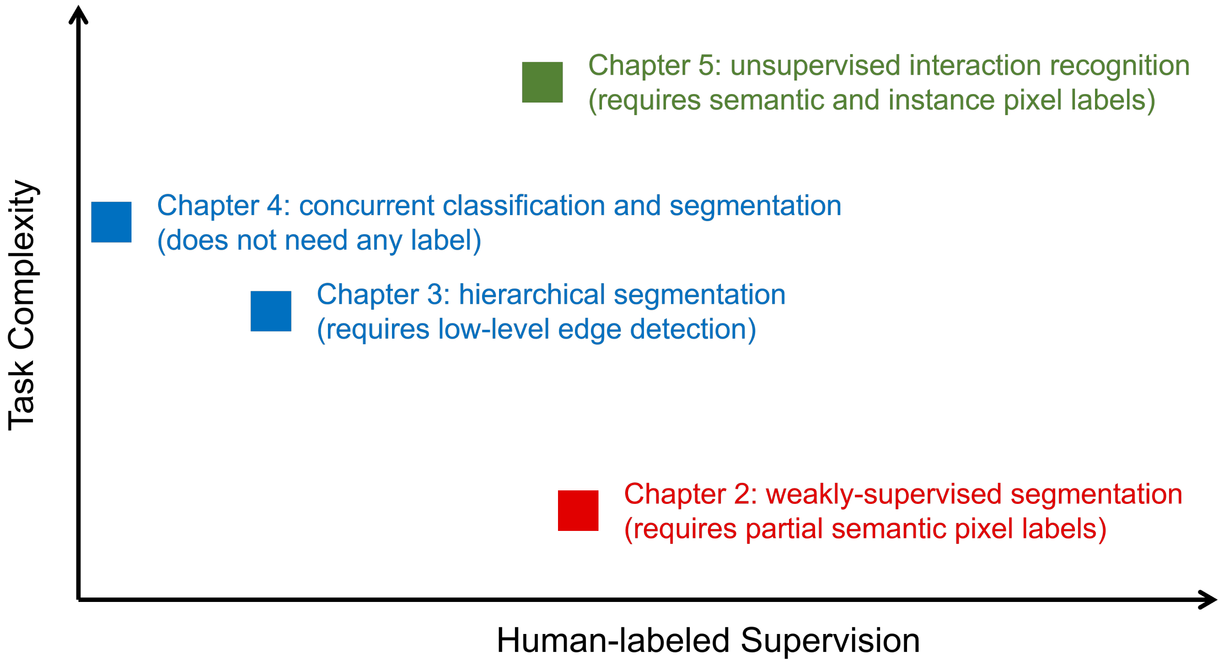

Chapter 5: unsupervised interaction recognition
(requires semantic and instance pixel labels)
Chapter 4: concurrent classification and segmentation
(does not need any label)
Chapter 3: hierarchical segmentation
(requires low-level edge detection)
Chapter 2: weakly-supervised segmentation
(requires partial semantic pixel labels)
Task Complexity
Human-labeled Supervision

Figure 1.8: We study four recognition perspectives to understand complex image scenes with minimal human labels. We organize these aspects based on the task complexity and the degree of human-labeled supervision. **Chapter 1:** We address weakly-supervised semantic segmentation, where partial semantic pixel labels are used. **Chapter 2:** We address unsupervised semantic segmentation, where only low-level edge detection is used. **Chapter 3:** We address unsupervised concurrent classification and segmentation in a single framework, where our model does not use any human labels. **Chapter 4:** We address unsupervised human-object interaction recognition. Semantic and instance pixels labels, no relationship labels, are used.

naturally inducing consistent multi-scale segmentations. Our model delivers both classification and hierarchical segmentations without the need for human-labeled supervision.

Unsupervised Human-Object Relationship Recognition

Chapter 5 introduces an unsupervised feature learning framework to tackle human-object relationship recognition [20, 19]. Existing methods consider it as a classification problem: the models predict action categories of a pair of human and object instances. Ground-truth labels are needed during training. Instead, we formulate relationship recognition as a feature learning problem: we group/separate objects based on their relationships with the surroundings. Our insight is to characterize such relationships with visual contexts, which emerge from discriminative feature learning. We learn features by contrasting pixels based on their semantics, instance ownerships, and surrounding spatial contexts. Our model recognizes human-object relationships without any supervision on relationships.

Summary

Chapter 6 summarizes the contributions of this dissertation. We aim at tackling image understanding of complex scenes with minimal human labels. We address the recognition problem from four perspectives. We organize these aspects based on the task complexity and the degree of human-labeled supervision in Fig. 1.8. From low to high levels of task complexity: **1)** We address weakly-supervised semantic segmentation, where partial semantic pixel labels are used. **2)** We address unsupervised semantic segmentation, where only low-level edge detections are used. **3)** We address unsupervised concurrent image classification and segmentation in a single framework, where our model does not use any human labels. **4)** We address unsupervised human-object recognition, where semantic and instance pixels labels, no relationship labels, are used. This dissertation explores more general and robust approaches to understanding the highly-complex and fast-changing real-world scenes.

Chapter 2

Universal Weakly Supervised Segmentation by Pixel-to-Segment Contrastive Learning

2.1 Introduction

Consider the task of learning a semantic segmenter given sparsely labeled training images (Fig. 2.1): Each body part is labeled with a single seed pixel and the task is to segment out the entire person by individual body parts, even though the ground-truth segmentation is not known during training. This task is challenging, as not only a single body part could contain several visually distinctive areas (*e.g., head* consists of *eyes*, *nose*, *mouth*, *beard*), but two adjacent body parts could also have the same visual appearance (*e.g., upper arm*, *lower arm*, and *hand* have the same skin appearance). Once the segmenter is learned, it can be applied to a test image without any annotations.

This task belongs to a family of weakly supervised segmentation problems, the goal of which is to assign a label to each pixel despite that only partial supervision is available during training. It addresses the practical issue of learning segmentation from minimum annotations. Such weak supervision takes many forms, e.g., image tags [38, 39, 40, 41], bounding boxes [42, 43, 35], keypoints [44], and scribbles [45, 46, 36]. Tags and boxes are coarse annotations that lack precise pixel localization whereas points and scribbles are sparse annotations that lack broad region coverage.

Weakly supervised semantic segmentation can be regarded as a semi-supervised pixel classification problem: Some pixels or pixel sets have labels, most don't, and the key is how to propagate and refine annotations from coarsely and sparsely labeled pixels to unlabeled pixels.

Existing methods tackle two types of weak supervision differently: Class Activation Maps (CAM) [37] are used to localize coarse labels, generate pseudo pixel-wise labels, and iteratively refine the segmentation model, whereas Conditional Random Fields (CRF) [47] are used to

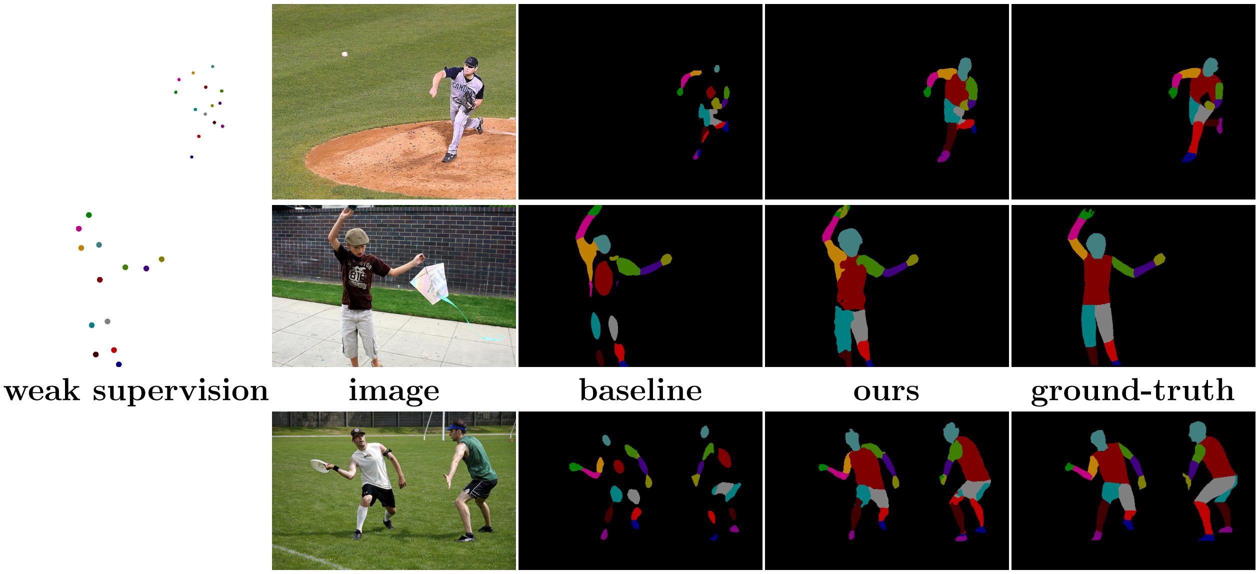


Figure 2.1: Our task learns a segmenter given partially labeled training images and applies it to test images. A common baseline is to propagate labels within an image based on feature similarity. We model it as semi-supervised metric learning and learn the pixel-wise feature by contrasting it within and across images. Our results are fuller and more accurate, approaching the ground-truth.

propagate sparse labels to the entire image.

These ideas can be incorporated as an additional unsupervised loss on the feature learned for segmentation [36]: While labeled pixels receive supervision, unlabeled pixels in different segments shall have distinctive feature representations.

We propose a *Semi-supervised Pixel-wise Metric Learning* (SPML) model that can handle all these weak supervision varieties with a single pixel-to-segment contrastive learning formulation (Fig. 2.2). Instead of classifying pixels, our metric learning model learns a pixel-wise feature embedding based on common grouping relationships that can be derived from any form of weak supervision.

Our key insight is to integrate unlabeled pixels into both supervised labeling and discriminative feature learning. They shall participate not only in data-driven grouping within each image, but also in discriminative feature learning *within* and more importantly *across* images. Intuitively, labeled pixels receive supervision not only for themselves, but also for their surround pixels that share visual similarity. On the other hand, unlabeled pixels are not just passively brought into discriminative learning induced by sparsely labeled pixels, they themselves are organized based on bottom-up grouping cues (such as grouping by color similarity and separation by strong contours). When they are examined *across* images, repeated patterns of frequent occurrences would also form a cluster that demand active discrimination from other patterns.

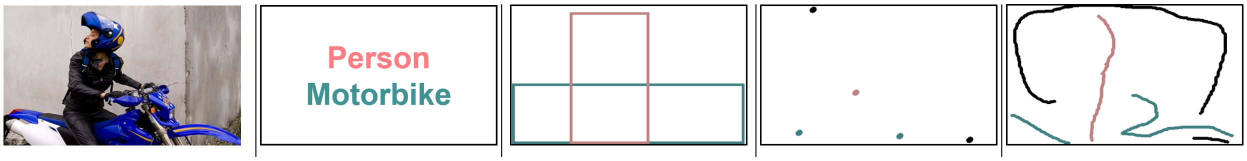


image	image tags	bounding boxes	labeled points	scribbles
SOTA methods	CAM + refine	box-wise CAM	CRF loss	CRF loss
our method	single pixel-to-segment contrastive learning loss formulation			
our relative gain	+8.6%	+4.7%	+24.7%	+1.4%

Figure 2.2: We propose a unified framework for weakly supervised semantic segmentation with different types of annotations. We demonstrate consistent performance gains compared to the SOTA methods: [34] for image tags, [35] for bounding boxes, and [36] for points and scribbles. For tags and boxes, Class Activation Maps (CAM) [37] are often used to localize semantics as an initial mask and iteratively refine the segmentation model, whereas for labeled points and scribbles, Conditional Random Fields (CRF) are used to propagate semantic labels to unlabeled regions based on low-level image similarity.

We capture the above insight in a single pixel-wise metric learning objective for segmentation, the goal of which is to map each pixel into a point in the feature space so that pixels in the same (different) semantic groups are close (far) in the feature space. Our model extends SegSort [48] from its fully supervised and unsupervised segmentation settings to a universal weakly-supervised segmentation setting. With a single consistent feature learning criterion, such a model sorts pixels discriminatively within individual images and sorts segment clusters discriminatively across images, both steps minimizing the same feature discrimination loss.

Our experiments on Pascal VOC [15] and DensePose [29] demonstrate consistent gains over the state-of-the-art (SOTA), and the gain is substantial especially for the sparsest keypoint supervision.

2.2 Related Work

Semi-supervised learning. [49] treats it as a joint learning problem with both labeled and unlabeled data. One way is to capture the underlying structure of unlabeled data with generative models [50, 51]. Another way is to regularize feature learning through a consistency loss, *e.g.,* adversarial ensembling [52], imitation learning and distillation [53], cross-view ensembling [54]. These methods are most related to transductive learning [55, 56, 57, 58], where labels are propagated to unlabeled data via clustering in the pre-trained feature space. Our work does transductive learning in an adaptively learned feature space.

Weakly-supervised semantic segmentation. Partial annotations include scribbles [45, 46,

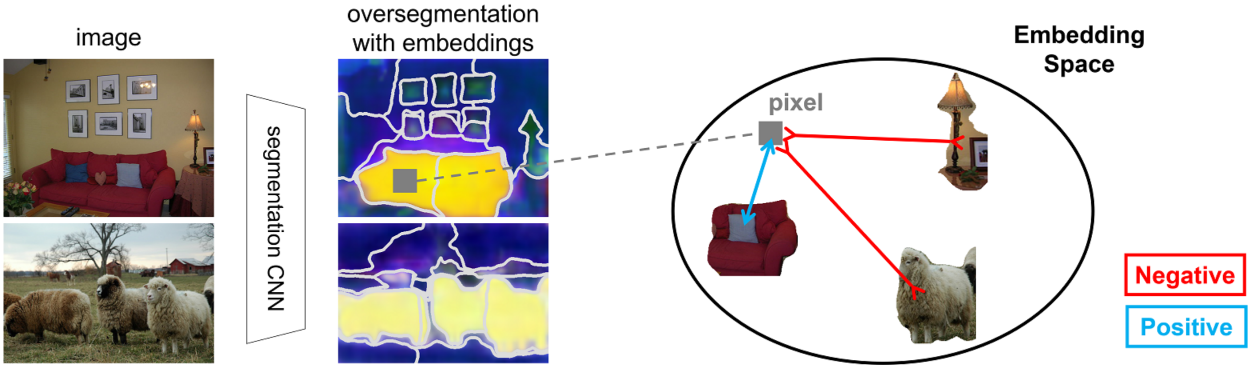


Figure 2.3: Overall method diagram. We develop pixel-wise embeddings with contrastive learning between pixels and segments. We derive various forms of positive and negative segments for each pixel. Our goal is to attract (blue inward arrows) the pixel with positive segments, while repelling (red outward arrows) it from negative segments in the feature space.

36, 59], bounding boxes [42, 43, 35], points [44], or image tags [60, 38, 39, 40, 61, 41, 62, 63, 64, 34, 65, 66, 67, 68]. [69] formulates all types of weak supervision as linear constraints on a SVM. [60] bootstraps segmentation predictions via EM-optimization. Recent works [45, 38, 70] typically use CAM [37] to obtain an initial dense mask and then train a model iteratively. GAIN [61] utilizes image tags or bounding boxes to refine these class-specific activation maps. [68] considers within-image relationships and explores the idea of co-segmentation. [67] estimates the foreground and background for each category, with which the network learns to generate more precise CAMs. Regularization is enforced at either the image level [45, 38, 70] or the feature level [46, 36] to produce better dense masks. We incorporate this concept into adaptive feature learning and train the model only once. All types of weak annotations are dealt with in a single contrastive learning framework.

Non-parametric segmentation. Prior to deep learning, non-parametric models [71, 72, 73] usually use designed features with statistical or graphical models to segment images. Recently, inspired by non-parametric models for recognition [74, 75], SegSort [48] captures pixel-to-segment relationships via a pixel-wise embedding and develops the first deep non-parametric semantic segmentation for supervised and unsupervised settings. Building upon SegSort, our work has the flexibility of a non-parametric model at capturing data relationships and modeling subclusters within a category.

2.3 Semi-Supervised Pixel-wise Metric Learning Method (SPML)

Metric learning develops a feature representation based on data grouping and separation cues. Our method (Fig. 2.3) segments an image by learning a pixel-wise embedding with a

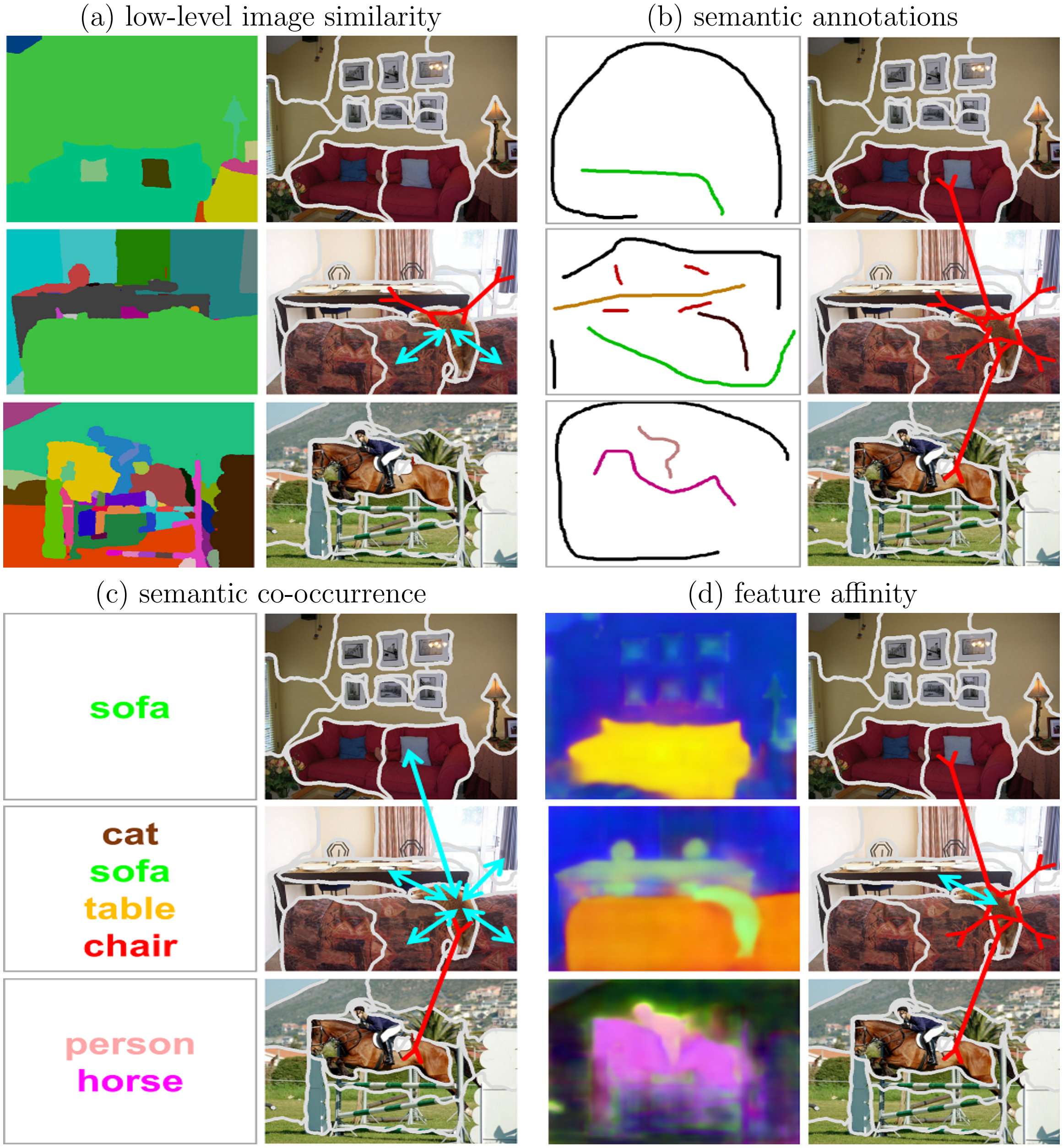


Figure 2.4: Four types of pixel-to-segment attraction and repulsion relationships. A pixel is attracted to (repelled by) segments: **a)** of similar (different) visual appearances such as color or texture, **b)** of the same (different) class labels, **c)** in images with common (distinctive) labels, **d)** of nearby (far-away) feature embeddings. They form different positive and negative sets. Using these visual relationships, we are able to include both labelled and unlabelled pixels / segments for disciminative feature learning.

contrastive loss between pixels and segments: For each pixel i, we learn a latent feature $\phi(i)$ such that i is close to its positive segments (exemplars) and far from its negative ones in that feature space.

In the fully supervised setting, we can define pixel i's positive and negative sets, denoted by $\mathcal{C}^+$ and $\mathcal{C}^-$ respectively, as pixels in the same (different) category. However, this idea is not applicable to weakly- or un-supervised settings where the label is not available on every pixel. In the labeled points setting, $\mathcal{C}^+$ and $\mathcal{C}^-$ would only contain a few exemplars according to the sparse pixel labels.

Our basic idea is to enlarge the sets of $\mathcal{C}^+$ and $\mathcal{C}^-$ to improve the feature learning efficacy. By exploring different relationships and assumptions in the image data, we are able to generate abundant positive and negative segments for any pixel at the same time, providing more supervision in the latent feature space. We propose four types of relationships between pixels and segments (Fig. 2.4):

1. **Low-level image similarity**: We impose a spatial smoothness prior on the pixel-wise feature to keep pixels together in visually coherent regions. The segment pixel i belongs to based on low-level image cues is a positive segment to pixel i; any other segments are negative ones.
2. **Semantic annotation**: We expand the semantics from labeled points and scribbles to pseudo-labels inferred from image- or box-wise CAM. The label of a segment can be estimated by majority vote among pixels; if it is the same as pixel i's, the segment is a positive segment to i.
3. **Semantic co-occurrence**: We expand the semantics by assuming that pixels in similar semantic contexts tend to be grouped together. If a segment appears in an image that shares any of the semantic classes as pixel i's image, it is a positive segment to i and otherwise a negative one.
4. **Feature affinity**: We impose a featural smoothness prior assuming that pixels and segments of the same semantics form a cluster in the feature space. We propagate the semantics within and across images from pixel i to its closest segment s in the feature space.

Pixel-to-Segment Contrastive Grouping Relationships

Our goal is to propagate known semantics from labeled data $\mathcal{C}$ to unlabeled data $\mathcal{U}$ with the aforementioned priors. $\mathcal{C}$ and $\mathcal{U}$ denote the sets of segment indices respectively. We detail how to augment positive / negative segment sets using both $\mathcal{C}$ and $\mathcal{U}$ for each type of relationships (Fig. 2.4).

Low-level image similarity. To propagate labels within visually coherent regions, we generate a low-level over-segmentation. Following SegSort [48], we use the HED contour detector [76] (pre-trained on BSDS500 dataset [77]) and gPb-owt-ucm [77] to generate a segmentation without semantic information. We define i's positive and negative segments as i's own segment and all the other segments, denoted as $\mathcal{V}^+$ and $\mathcal{V}^-$ respectively. We only consider segments in the same image as pixel i's. We align the contour-based over-segmentations with segmentations generated by K-Means clustering as in SegSort.

Semantic annotation. Image tags and bounding boxes do not provide pixel-wise localization. We derive pseudo labels from image- or box-wise CAM and align them with oversegmentations induced by the pixel-wise feature. Pixel i's positive (negative) segments are the ones with the same (different) semantic category, denoted by $\mathcal{C}^+$ and $\mathcal{C}^-$ respectively. We ignore all the unlabeled segments.

Semantic co-occurrence. Semantic context characterizes the co-occurrences of different objects, which can be used as a prior to group and separate pixels. We define semantic context as the union of object classes in each image. Even without the pixel-wise localization of semantic labels, we can leverage semantic context to impose global regularization on the latent feature: The feature should separate images without any overlapping object categories.

Let $\mathcal{O}^+$ ($\mathcal{O}^-$) denote the set of segments in images with (without) overlapping categories as pixel i's image. That is, if the image of pixel i and another image share any semantic labels (Fig. 2.4c: {*cat, sofa, table, chair*} for the pixel in the Row 2 image vs. {*sofa*} for the Row 1 image), then all the segments from that image are positive segments to i and included in $\mathcal{O}^+$; otherwise they are considered negative segments in $\mathcal{O}^-$ (Fig. 2.4c: all the segments in the Row 3 image). In particular, all the segments in pixel i's image are in $\mathcal{O}^+$ of i. This semantic context relationship does not require localized annotations yet imposes regularization on pixel feature learning.

Feature affinity. Our goal is to learn a pixel-wise feature that indicates semantic segmentation. It is thus reasonable to assume that pixels and segments of the same semantics form a cluster in the feature space, and we reinforce such clusters with a featural smoothness prior: We find nearest neighbours in the feature space and propagate labels accordingly.

Specifically, we assign a semantic label to each unlabeled segment by finding its nearest labeled segment in the feature space. We denote this expanded labeled set by $\hat{\mathcal{C}}$. For pixel i, we define its positive (negative) segment set $\hat{\mathcal{C}}^+$ ($\hat{\mathcal{C}}^-$) according to whether a segment has the same label as i.

Our feature affinity relationship works best when: 1) the original labeled set is large enough to cover the feature space, 2) the labeled segments are distributed uniformly in the feature space, and 3) the pixel-wise feature already encodes certain semantic information. We thus only apply to DensePose keypoint annotations in our experiments, where each body part is annotated by a point.

Pixel-wise Metric Learning Loss

SegSort [48] is an end-to-end segmentation model that generates a pixel-wise feature map and a resulting segmentation. Assuming independent normal distributions for individual segments, SegSort seeks a maximum likelihood estimation of the feature mapping, so that the feature induced partitioning in the image and clustering across images provide maximum

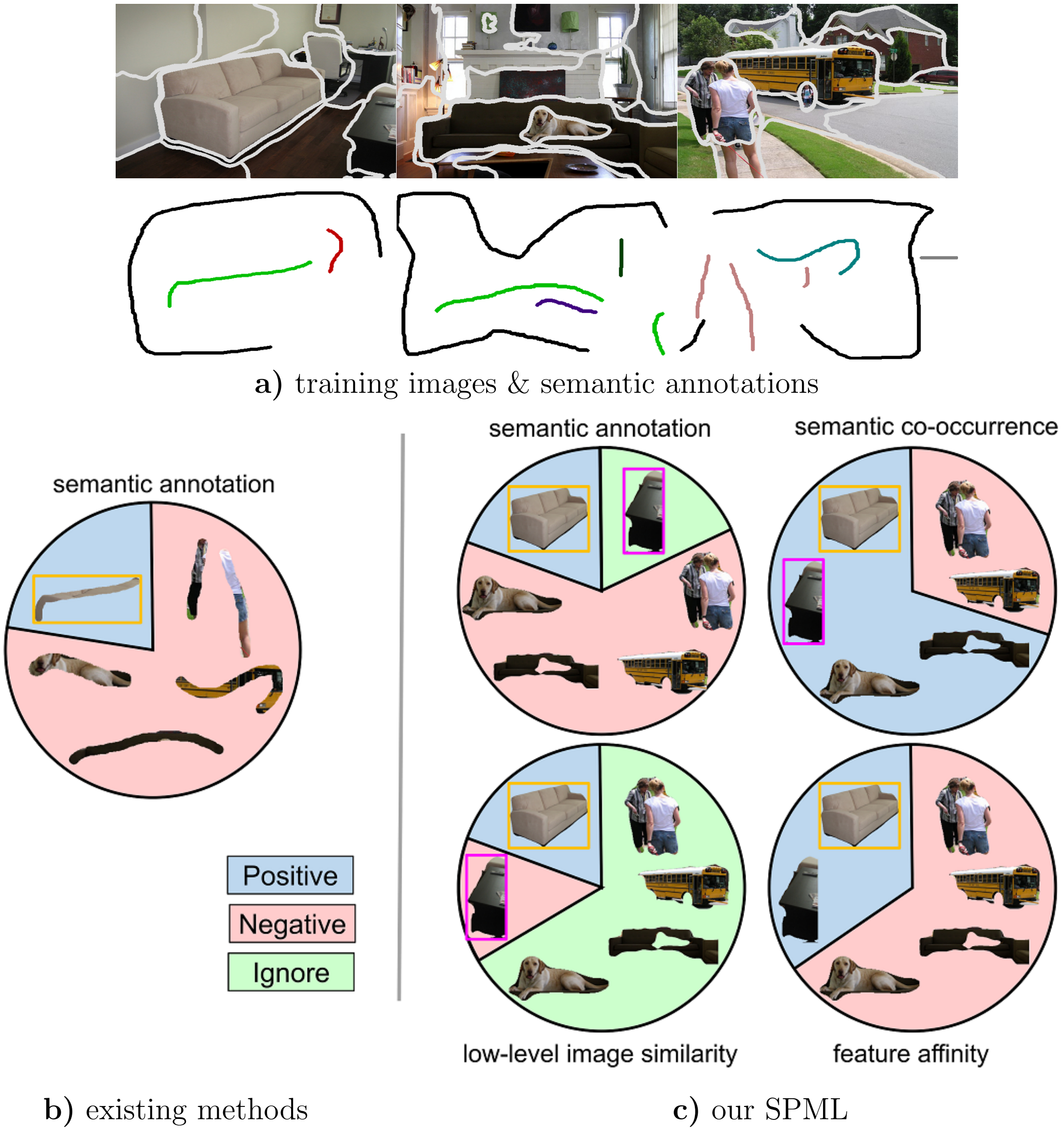


Figure 2.5: Our method uses labeled and unlabeled portions of the training data more extensively. **a)** Training images and their labeled scribbles are sparse and incomplete. **b)** Existing methods train a pixel-wise classifier using only labeled pixels and propagate labels within each image. **c)** Our method leverages *four* types of pixel-to-segment semantic relationships to augment the labeled sets, includes unlabeled pixels (fuller segments than just thin scribbles) and unlabeled segments (e.g. *desk* outlined in magenta), forms dynamic contrastive relationships between segments (e.g. the *desk* can be positive, negative, or to be ignored to the *sofa* in different relations.

discrimination among segments. During inference, the segment label is predicted by K-Nearest Neighbor retrievals.

The feature induced partitioning in each image is calculated via spherical K-Means clustering [78]. Let $\boldsymbol{e}_i$ denote the feature vector at pixel i, which contains the mapped feature $\phi(i)$ and i's spatial coordinates. Let z_i denote the index of the segment that pixel i belongs to, $\boldsymbol{R}_s$ the set of pixels in segment s, and $\boldsymbol{\mu}_s$ the segment feature calculated as the spherical cluster centroid of segment s. In the Expectation-Maximization (EM) procedure for spherical K-means, the E-step calculates the most likely segment pixel i belongs to: $z_i = \text{argmax}_s\, \boldsymbol{\mu}_s' \boldsymbol{e}_i$, and the M-Step updates the segment feature as the mean pixel-wise feature: $\boldsymbol{\mu}_s = \frac{\sum_{i \in \boldsymbol{R}_s} \boldsymbol{e}_i}{\| \sum_{i \in \boldsymbol{R}_s} \boldsymbol{e}_i \|}$.

Let s denote the resulting segment that pixel i belongs to per spherical clustering. The posterior probability of pixel i in segment s can be evaluated over the set of all segments S as:

$$p(z_i = s | \boldsymbol{e}_i, \boldsymbol{\mu}) = \frac{\exp(\kappa\, \boldsymbol{\mu}_s' \, \boldsymbol{e}_i)}{\sum_{t \in S} \exp(\kappa\, \boldsymbol{\mu}_t' \, \boldsymbol{e}_i)} \tag{2.1}$$

where κ is a concentration hyper-parameter. SegSort minimizes the negative log-likelihood loss:

$$L_{\text{SegSort}}(i) = -\log p(z_i = s | \boldsymbol{e}_i, \boldsymbol{\mu}) = -\log \frac{\exp(\kappa\, \boldsymbol{\mu}_s' \, \boldsymbol{e}_i)}{\sum_{t \in S} \exp(\kappa\, \boldsymbol{\mu}_t' \, \boldsymbol{e}_i)}. \tag{2.2}$$

SegSort adopts soft neighborhood assignment [79] to further strengthen the grouping of same-category segments. Let $\mathcal{C}^+$ ($\mathcal{C}^-$) denote the index set of segments in the same (different) category as pixel i except s – the segment i belongs to. We have:

$$L_{\text{SegSort}^+}(i, \mathcal{C}^+, \mathcal{C}^-) = -\log \sum_{t \in \mathcal{C}^+} p(z_i = t | \boldsymbol{e}_i, \boldsymbol{\mu}) = -\log \frac{\sum_{t \in \mathcal{C}^+} \exp(\kappa\, \boldsymbol{\mu}_t' \, \boldsymbol{e}_i)}{\sum_{t \in \mathcal{C}^+ \cup \mathcal{C}^-} \exp(\kappa\, \boldsymbol{\mu}_t' \, \boldsymbol{e}_i)}. \tag{2.3}$$

For our weakly supervised segmentation, the total pixel-to-segment contrastive loss for pixel i consists of 4 terms, one for each of the 4 pixel-to-segment attraction and repulsion relationships:

$$\begin{aligned} L(i) = {} & \lambda_I L_{\text{SegSort}^+}(i, \mathcal{V}^+, \mathcal{V}^-) + \lambda_C L_{\text{SegSort}^+}(i, \mathcal{C}^+, \mathcal{C}^-) \\ & + \lambda_O L_{\text{SegSort}^+}(i, \mathcal{O}^+, \mathcal{O}^-) + \lambda_A L_{\text{SegSort}^+}(i, \hat{\mathcal{C}}^+, \hat{\mathcal{C}}^-), \end{aligned} \tag{2.4}$$

where $\lambda_C = 1$. Fig. 2.5 shows how our metric learning method utilizes labeled and unlabeled pixels and segments more extensively than existing classification methods: Our pseudo-labeled sets are fuller than labeled thin scribbles and include unlabeled segments; there are 3 more relationships other than semantic annotations; our segments participate in contrastive learning with dynamic roles in different relations. By easily integrating a full range of pixel-to-segment attraction and repulsion relationships from low-level image similarity to mid-level feature affinity, and to high-level semantic co-occurrence, we go far beyond the direct supervision from semantic annotations.

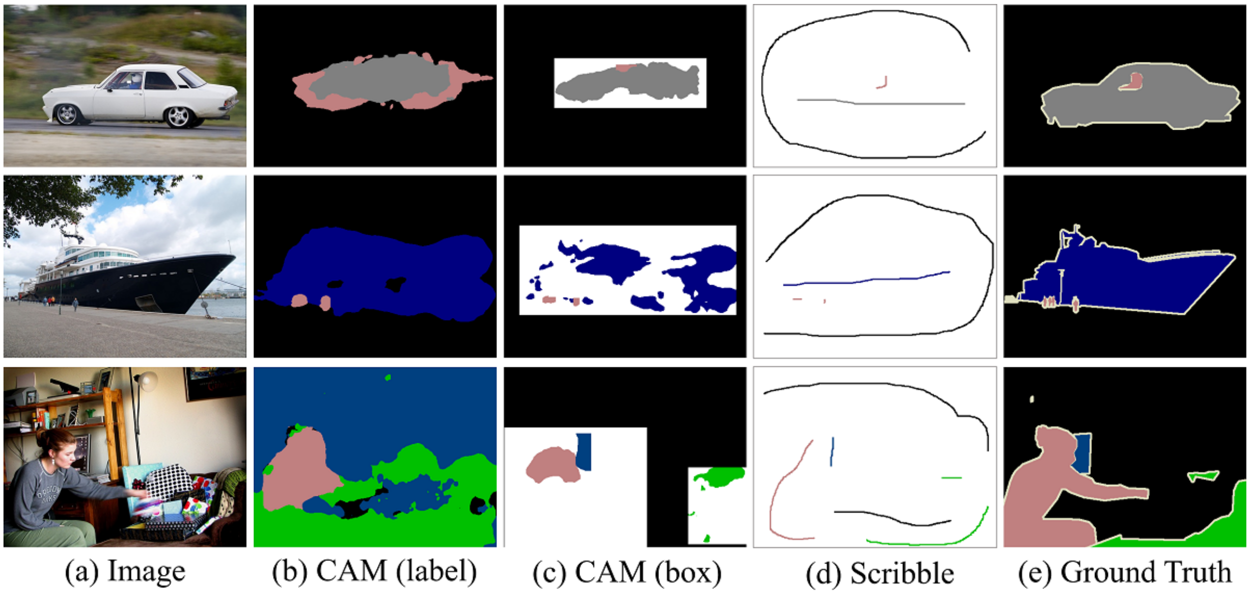


Figure 2.6: Visual examples of semantic annotations used on VOC. For image tag and bounding box annotation, we use the classifier trained by [66] to infer CAM as semantic annotation. These semantic annotations are noisy, which do not precisely localize on the objects.

2.4 Experiments

Datasets

We conduct extensive experiments over Pascal VOC 2012 and Densepose datasets using different forms of weak annotations.

Pascal VOC 2012 [15] includes 20 object categories and one background class. Following [80], we use the augmented training set with 10,582 images and validation set with 1,449 images. We use the scribble annotations provided by [45] for training.

DensePose [29] is a human pose parsing dataset based on MSCOCO [16]. The dataset is annotated with 14 body part classes. We extract the keypoints from the center of each part segmentation. The training set includes 26,437 images and we use minival2014 set for testing, which includes 1,508 images.

Weak Annotations on Pascal VOC 2012

Since image tag and bounding box annotations do not provide any of precisely localized semantic information, we adopt CAM [37] to produce localized semantic cues. Without using additional saliency labels, we use the classifier trained by [66] to generate CAM. Let $\mathcal{M}_c$ be

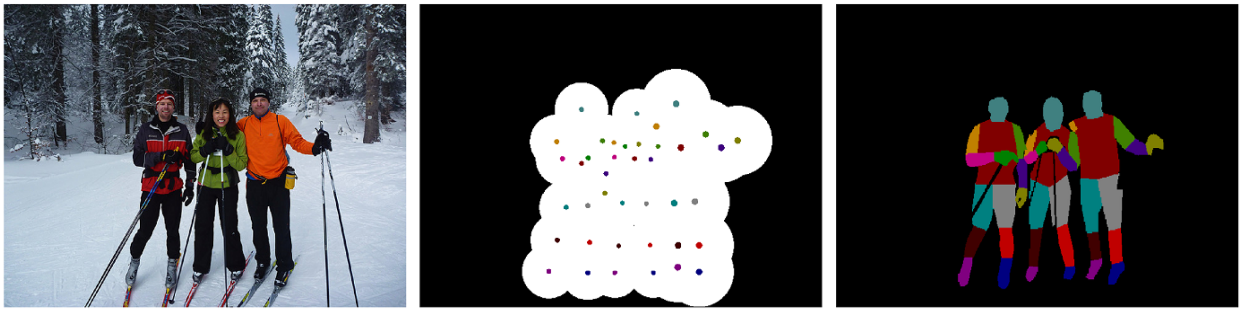


Figure 2.7: Preparing training labels on DensePose dataset. From left to right are input image, our training labels and ground-truth mask. For each keypoint, a Gaussian heat map is applied to determine labelled, unknown and background region. The white region denotes unknown pixels, to which we propagate labels from annotated or background region.

the activation map of class c.

For image tag annotations, we follow [39] to normalize $\mathcal{M}_c$ of the entire image within the range between 0 and 1, where $\mathcal{M}_c = \frac{\mathcal{M}_c}{\max_c \mathcal{M}_c}$. The background confidence $\mathcal{M}_{bg}$ can then be estimated by $\mathcal{M}_{bg} = (1 - \max_c \mathcal{M}_c)^{\alpha}$, where α is the hyper-parameter adjusting background confidence. In our experiments, we set α to 6 and confidence threshold to 0.2. The low-confidence pixels are considered as unlabeled regions.

For bounding box annotations, we simply normalize the CAM logits within each bounding box to the range between 0 and 1. We then set confidence threshold to 0.5 for selecting foreground pixels and unlabeled regions. We restrict all the regions outside bounding boxes as "background". See figure 2.6 for more visual examples.

Data pre-processing for DensePose dataset

We next illustrate our pre-processing to generate training labels given keypoint annotations in DensePose dataset. As shown in figure 2.7, we first assume a Gaussian heat map from every keypoint. By thresholding, we derive 3 regions from every Gaussian blob: labelled, unknown and background region. In labelled region, pixels are annotated as each body part. We then propagate labels, including background class, to pixels in the unknown region. The *std* of Gaussian heat map is estimated from instance size, and we use ground-truth information in our paper.

Architecture and Training

For all the experiments on PASCAL VOC, we base our architecture on DeepLab [80] with ResNet101 [81] as the backbone network. For the experiments on DensePose, we adopt PSPNet [82] as the backbone network. Our models are pre-trained on ImageNet [11] dataset.

Dataset	Annotation	λ_I	κ_I	λ_C	κ_C	λ_O	κ_O	λ_A	κ_A	batchsize
VOC	scribbles	0.1	16	1.0	6	0.5	12	0.0	-	12
	points	1.0	16	1.0	6	1.0	8	0.0	-	12
	boxes	0.3	16	1.0	6	1.0	8	0.0	-	16
	image tags	0.3	16	1.0	6	1.0	8	0.0	-	16
DensePose	points	0.1	16	1.0	6	0.0	-	0.5	12	16

Table 2.1: Hyper-parameters for different types of annotations on Pascal and DensePose dataset.

For each type of annotations and dataset, we formulate four types of pixel-to-segment contrastive relationships and jointly optimize them in a single pixel-wise metric learning framework (Fig. 2.3).

We next describe the hyper-parameters used for each experiment. On Pascal VOC dataset, we set "batchsize" to 12 and 16 for scribble / point and image tag / bounding box annotations. On DensePose dataset, "batchsize" is set to 16. For all the experiments, we train our models with 512 × 512 "cropsize". Following [80], we adopt poly learning rate policy by multiplying base learning rate by $1 - (\frac{iter}{max\_iter})^{0.9}$. We set initial learning rate to 0.003, momentum to 0.9. For the hyper-parameters in SegSort framework, we use unit-length normalized embedding of dimension 64 and 32 on VOC and DensePose, respectively. We iterate K-Means clustering for 10 iterations and generate 36 and 144 clusters on VOC and DensePose dataset. We set the concentration parameter κ to different values for **semantic annotation**, **low-level image similarity**, **semantic co-occurrence** and **feature affinity**, respectively. Moreover, λ_I, λ_O and λ_A are set to different values according to different types of annotations and datasets. λ_C is set to 1 among all the experiments. The detailed hyper-parameter settings are summarized in table 2.1. We train for $30k$ and $45k$ iterations on VOC and DensePose dataset for all the experiments. We use additional memory banks to cache up previous 2 batches. For conducting experiments, we take advantage of XSEDE infrastructure [83] that includes Bridges resources [84].

Inference and Testing

We fix the learned pixel-wise embedding and train an additional softmax classifier for inference. Iterative training is adopted to bootstrap the semantic segmentation prediction. Notably, we do not propagate gradients to the segmentation CNN from the softmax classifier.

For scribbles / points / bounding boxes, we first learn an initial softmax classifier S_1 from the corresponding weak annotations. Following [39], we apply random walk to refine the semantic logits $\tilde{\mathcal{M}}$ generated by S_1. The transition probability matrix T is formulated

Algorithm 1: Inference procedure for semantic segmentation using scribble / point / bounding box annotations.

```
Input: Fixed pixel-wise embedding $\boldsymbol{e}$ of the input image and weak annotations $\mathcal{Y}_{weak}$.
Output: Semantic segmentation prediction $\mathcal{Y}_{pred}$.
/* Train the initial softmax classifier */
Train the softmax classifier $S_1$ using $\mathcal{Y}_{weak}$.
/* Train the final softmax classifier */
Predict semantic logits from initial softmax classifier: $\tilde{\mathcal{M}} = S_1(\boldsymbol{e})$.
Calculate pixel-wise transition probability matrix $T$ from $\boldsymbol{e}$.
Refine semantic logits by random walk propagation: $\tilde{\mathcal{M}}' = T^\top \circ ... \circ T^\top \tilde{\mathcal{M}}$.
Derive pseudo labels from refined semantic logits: $\mathcal{Y}_{sc} = \text{argmax}_c \tilde{\mathcal{M}}'_c$.
Train the softmax classifier $S_2$ using $\mathcal{Y}_{sc}$.
Predict final semantic segmentation $\mathcal{Y}_{pred}$ from $S_2$.
```

Algorithm 2: Inference procedure for semantic segmentation using image-level tags.

```
Input: Fixed pixel-wise embedding $\boldsymbol{e}$ of the input image and CAM logits $\mathcal{M}$.
Output: Semantic segmentation prediction $\mathcal{Y}_{pred}$.
/* Train the initial softmax classifier */
Calculate pixel-wise transition probability matrix $T$ from $\boldsymbol{e}$.
Refine CAM by random walk propagation: $\mathcal{M}' = T^\top \circ ... \circ T^\top \mathcal{M}$.
Derive pseudo labels from refined CAM: $\mathcal{Y}^1_{cam} = \text{argmax}_c \mathcal{M}'_c$.
Predict new pseudo labels $\mathcal{Y}^1_{nn}$ from $\mathcal{Y}^1_{cam}$ using nearest neighbor retrievals.
Train the softmax classifier $S_1$ using $\mathcal{Y}^1_{nn}$.
/* Train the final softmax classifier */
Predict pseudo labels $\mathcal{Y}^2_{sc}$ from initial softmax classifier $S_1$.
Predict new pseudo labels $\mathcal{Y}^2_{nn}$ from $\mathcal{Y}^2_{sc}$ using nearest neighbor retrievals.
Train the softmax classifier $S_2$ using $\mathcal{Y}^2_{nn}$.
Predict final semantic segmentation $\mathcal{Y}_{pred}$ from $S_2$.
```

as follows: $T_{i,j} = (\frac{\exp(\gamma \boldsymbol{e}_i^\top \boldsymbol{e}_j)}{\sum_j \exp(\gamma \boldsymbol{e}_i^\top \boldsymbol{e}_j)})^\beta$, where β and γ are 20 and 5, respectively. The label propagation is given by: $\tilde{\mathcal{M}}' = T^\top \tilde{\mathcal{M}}$, where $\tilde{\mathcal{M}}'$ denotes refined semantic logits. The random walk process is iterated for 6 times. Next, we obtain the corresponding pseudo labels $\mathcal{Y}_{sc} = \text{argmax}_c \tilde{\mathcal{M}}'_c$. The pseudo labels are used to train the final softmax classifier S_2 for predicting semantic segmentation.

For image tag annotations, we adopt both within-image and across-image label propagation to generate optimal pseudo labels. Starting with CAM logits $\mathcal{M}$, we conduct within-image label propagation thru random walk and obtain refined pseudo labels $\mathcal{Y}^1_{cam}$. Across-image label propagation is carried out by nearest neighbor search thru the whole training set. We refer to SegSort [48] for more details. We then obtain refined pseudo labels $\mathcal{Y}^1_{nn}$ and train the initial softmax classifier S_1. Similarly, we use S_1 to predict pseudo labels $\mathcal{Y}_{sc}^2$ from the training images. Followed by nearest neighbor search, we obtain our final pseudo labels $\mathcal{Y}_{nn}^2$ and train the final semantic classifier S_2.

The inference procedures for different annotations are summarized in algorithm 1 and 2, respectively. For image tags, we adopt multi-scale and horizontally flipping as data augmentation for predicting semantic segmentation. For scribbles / points / bounding boxes, we do not employ data augmentation during the final inference.

Pascal: Image tags	Saliency	*val*	*test*
DSRG [40]	✓	61.4	63.2
FickleNet [41]	✓	64.9	65.3
RRM [63]	-	66.3	66.5
SGAN [64]	✓	67.1	67.2
SCE [34]	-	66.1	65.9
Our SPML	-	**69.5**	**71.6**

Pascal: Bounding boxes	*val*	*test*
SDI [43]	69.4	-
BCM [35]	70.2	-
Our SPML	**73.5**	**74.7**

Table 2.2: Pascal VOC 2012 dataset with image tag (left) and bounding box (right) annotations.

Quantitative Results on Pascal VOC 2012 Dataset

We demonstrate the superior efficacy of our method using over all types of weak annotations on Pascal VOC 2012 datataset.

We first report performance on VOC validation set. **1) image tag annotations:** Table 2.2) shows that, without using additional saliency labels, our method outperforms existing methods with saliency by 4.4%, and those without saliency by 5.1%. **2) bounding box annotations:** Table 2.2 shows that, with the same DeepLab/ResNet101 backbone network, our method outperforms existing methods by 3.2%. **3) scribble and point annotations:** Table 2.3 shows that, our method consistently delivers the best performance among methods without or with CRF post-processing. We get 74.2% (76.1%) mIoU, achieving 97.5% (98.4%) of full supervision performance in these two categories respectively.

We next report per-category results on Pascal VOC. In table 2.4, we compare with [36] on VOC validation set using scribble annotations. Without- and with CRF post-processing, our method outperform the baseline method among most categories by large margin. We further conduct experiments on VOC testing set, using DeepLab as backbone network. In table 2.5, we can retrieve most performance w.r.t full supervision.

Lastly, we demonstrate the efficacy of our method by varying sparsity of scribble and point annotations. Exploiting metric learning with different relationships in the data frees us from the classification framework and delivers a more powerful approach that requires fewer annotations. Table 2.3 shows that, as we shorten the length of scribbles from 100%, 80%, 50%, 30% to

Pascal: Scribbles	CRF	Full	Weak	WvF
NormalCut [46]		75.6	72.8	96.3
NormalCut [46]	✓	76.8	74.5	97.0
KernelCut [36]		75.6	73.0	96.6
KernelCut [36]	✓	76.8	75.0	97.7
BPG [59]		75.6	73.2	96.8
BPG [59]	✓	76.8	76.0	99.0
Our SPML		76.1	**74.2**	**97.5**
Our SPML	✓	77.3	76.1	98.4

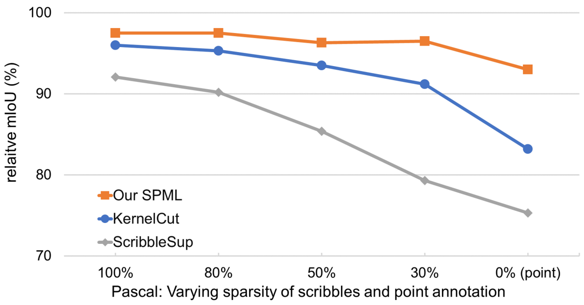


Table 2.3: Pascal VOC 2012 dataset using scribble annotations. **Left**: mIoU on validataion set. **WvF** denotes relative mIoU w.r.t full supervision. **Right**: Relative mIoU performance w.r.t full supervision on different lengths of scribbles.

Backbone	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
KernelCut [36]	83.2	35.8	82.8	66.8	75.1	90.9	83.9	89.2	35.8	82.5	53.7	83.4	83.2	79.5	82.2	57.6	81.9	41.6	81.1	73.5	73.2
Our SPML	**85.8**	**37.6**	82.8	**69.6**	**75.9**	89.3	82.8	**89.7**	**38.6**	**85.7**	**56.7**	**85.9**	80.1	78.1	**84.8**	53.9	**83.7**	**49.2**	80.9	**74.4**	**74.2**
KernelCut [36]	86.2	37.3	85.5	69.4	77.8	91.7	85.1	91.2	38.8	85.1	55.5	85.6	85.8	81.7	84.1	61.4	84.3	43.1	81.4	74.2	75.2
Our SPML	**89.0**	**38.4**	**86.0**	**72.6**	**77.9**	90.0	83.9	91.0	**40.0**	**88.3**	**57.7**	**87.7**	82.8	79.1	**86.5**	57.1	**87.4**	**50.5**	81.2	**76.9**	**76.1**

Table 2.4: Per-class results on Pascal VOC 2012 validation set. White- and gray-colored background denotes using without- and with- CRF post-processing for inference.

Annotations	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
Full mask	91.5	43.5	83.0	67.9	81.7	89.8	88.7	94.6	37.5	81.6	68.7	88.8	82.4	88.6	87.6	64.1	87.6	52.7	76.5	71.4	77.3
Scribbles	87.0	36.7	82.3	65.5	79.7	89.5	84.8	90.1	37.6	86.3	63.1	89.1	87.8	83.0	86.0	65.8	85.8	60.3	76.9	73.0	76.4
Points	83.5	37.0	78.4	61.9	74.8	86.4	83.2	86.9	37.9	85.3	62.4	87.2	84.2	81.1	83.1	64.3	85.1	59.1	74.0	66.3	74.0
Boxes	84.1	36.5	86.7	57.6	75.7	87.7	84.8	89.6	39.4	86.4	57.2	89.2	88.0	82.6	80.3	54.7	88.2	55.9	79.7	71.6	74.7
Tags	82.1	38.7	80.0	56.9	73.7	85.7	81.0	86.7	33.9	87.7	60.8	86.8	84.9	81.3	77.7	53.2	86.5	50.1	64.8	58.4	71.6

Table 2.5: Per-class results on Pascal VOC 2012 testing set. CRF post-processing is used for inference.

0% (points), we reach $97.5\%, 97.5\%, 96.3\%, 96.5\%$ and 93.7% of full supervision performance. Compared to the full scribble annotations, our accuracy only drops 3.7% with point labels and is significantly better than the baseline. We report absolute mIoU performance by varying sparsity of scribbles on Pascal VOC 2012 validation set. The results are summarized in table 2.6. Our results are much better with sparser annotation.

Quantitative Results on DensePose Dataset

We adopt point annotations for training on DensePose dataset. For comparison, we train our baseline using the code released by [36]. Table 2.7 shows that, our method without CRF post-processing outperforms the baseline by 12.9% mIoU, reaching 77.1% of full supervision

Method	Backbone	CRF	Full	100%	80%	50%	30%	0%
ScribbleSup [45]	DeepLab-MSc-LargeFOV	✓	68.5	63.1	61.8	58.5	54.3	51.6
KernelCut [36]	DeepLab-MSc-LargeFOV	✓	68.7	66.0	65.5	64.2	62.7	57.2
Our SPML	DeepLab/ResNet101		76.1	74.2	74.2	73.3	73.4	71.3
Our SPML	DeepLab/ResNet101	✓	77.3	76.1	75.8	74.8	75.0	73.2

Table 2.6: mIoU performance on Pascal VOC 2012 validation set on different lengths of scribble.

performance with only point supervision. We outperform the baseline method by large margin in every category.

Method	bg.	torso	RHand	LHand	LFoot	RFoot	RThigh	LThigh	RLeg	LLeg	LArm	RArm	LFarm	RFarm	Heaad	mIoU	WvF
Softmax	96.2	73.7	61.1	57.2	37.2	37.8	56.8	54.8	49.7	49.5	62.0	63.8	58.3	61.5	84.6	60.3	-
SegSort	95.8	71.9	57.4	53.0	33.4	33.4	54.0	51.8	46.4	46.9	59.2	61.1	54.4	57.9	83.2	57.3	-
KernelCut [36]	87.2	28.3	37.5	36.0	18.9	19.5	21.2	20.8	16.1	16.6	33.9	35.3	35.6	37.6	25.2	31.3	51.9
Our SPML	93.8	57.7	48.1	43.2	22.8	22.2	36.6	35.6	27.1	27.6	42.1	45.3	42.0	45.5	72.6	44.2	77.1

Table 2.7: Per-class results on DensePose minival 2014 set with keypoint annotations. White- and gray-colored background indicates using full and point supervision.

Ablation Study of Hyper-parameters

We conduct ablation study over different regularizations on Pascal VOC dataset. As shown in Table 2.8, we achieve the most optimal performance on Pascal VOC dataset with $\lambda_I = 0.1$ and $\lambda_O = 0.5$. We also observe performance drops 0.4 of mIoU by adding **feature affinity** regularization. We argue that scribble/box/point annotations are not uniformly distributed across object instance and background, and results in noisy label propagation.

Visual Results on Pascal VOC and DensePose Dataset

We present the visual results on VOC (with image tags, bounding boxes and scribbles) and DensePose (with keypoints) dataset in figure 2.8. We observe that our segmentation results are better aligned with image boundary. When visual evidence is prominent, our weakly-supervised results are even better than the fully-supervised counterpart. We then demonstrate the efficacy of each visual relationship in figure 2.9. By adding **semantic annotation**, **low-level image similarity** and **feature affinity** progressively, we observe consistent improvement of our results. The predicted segmentation becomes more coherent and better aligned with image boundary.

λ_I	λ_O	mIoU
0.3	0.5	73.7
0.1	0.5	74.2
0.05	0.5	73.5
0	0.5	71.7

λ_I	λ_O	mIoU
0.1	1.0	74.1
0.1	0.5	74.2
0.1	0.1	74.1
0.1	0	72.8

λ_I	λ_O	λ_A	mIoU
0	0	0	71.2
0.1	0	0	72.8
0.1	0.5	0	74.2
0.1	0.5	0.1	73.8

Table 2.8: Ablation study of different weighting parameters for each objective function on Pascal VOC validation dataset.

Visual Results on Nearest Neighbor Segment Retrievals

We lastly showcase that our method implicitly encodes semantic contexts. In Fig. 2.10, We observe that retrieved segments appear in the similar semantic context as the query segments. For examples, given a bottle next to a desktop, our model retrieves bottles also next to a desktop; a set of sofas in a living room can be retrieved using one sofa query example; screens of a desktop can also be retrieved likewise.

2.5 Summary

We propose a novel weakly-supervised semantic segmentation method via Semi-supervised Pixel-wise Metric Learning, based on four common types of pixel-to-segment attraction and repulsion relationships. It is universally applicable to various weak supervision settings, whether the training images are coarsely annotated by image tags or bounding boxes, or sparsely annotated by keypoints or scribbles. Our results on PASCAL VOC and DensePose show consistent and substantial gains over SOTA, especially for the sparsest keypoint supervision.

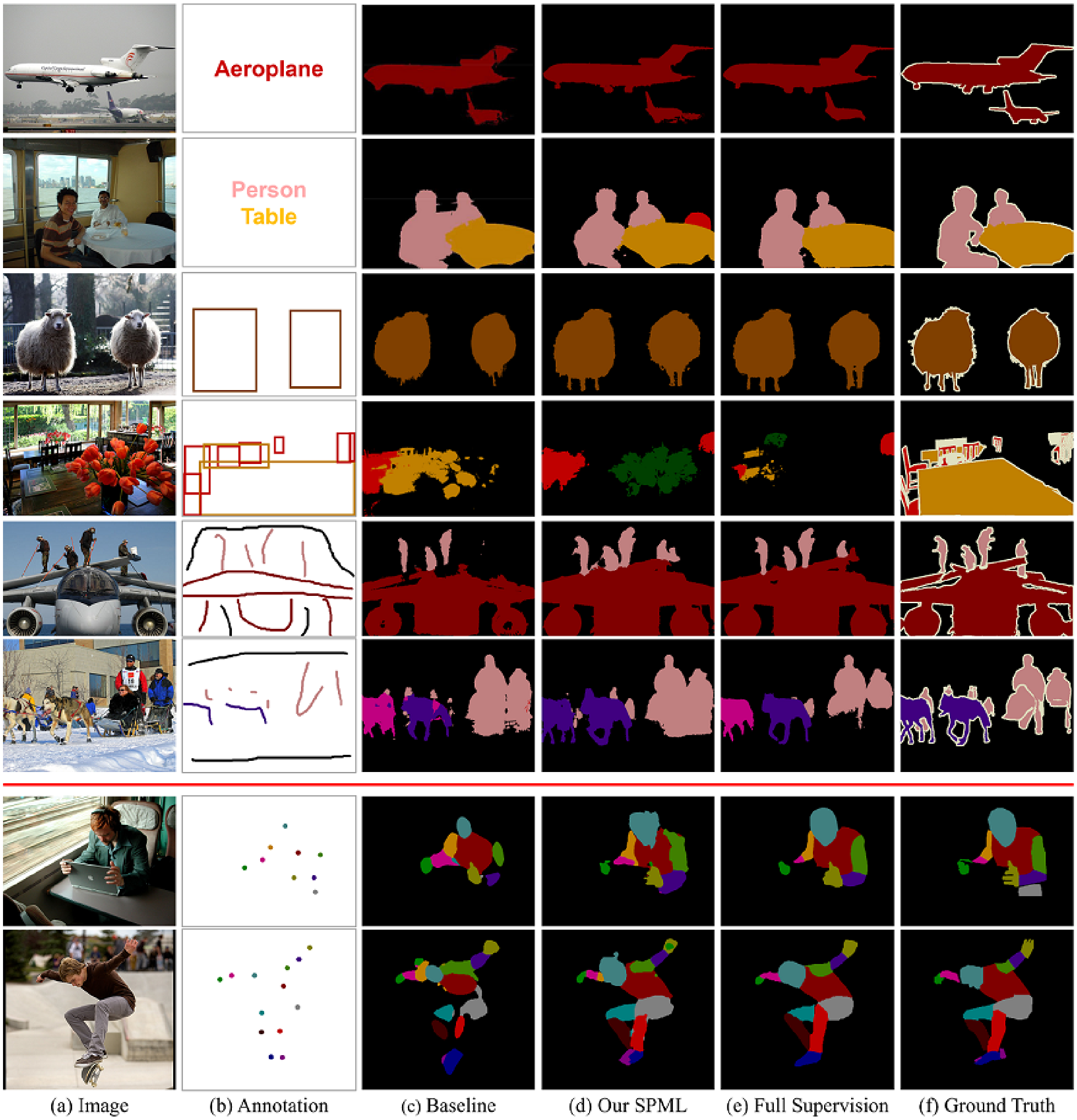


Figure 2.8: Visual comparison of baseline method (c), our SPML (d) and fully-supervised SegSort (e) on VOC and DensePose. On VOC (top 6 rows), our baseline method is based on [41, 35, 36] for image tag, bounding box and scribble annotations, respectively. On DensePose (bottom 2 rows), our baseline is [36]. The results from our weakly-supervised model is visually very close to its fully-supervised counterpart, or even better when visual cues are prominent.

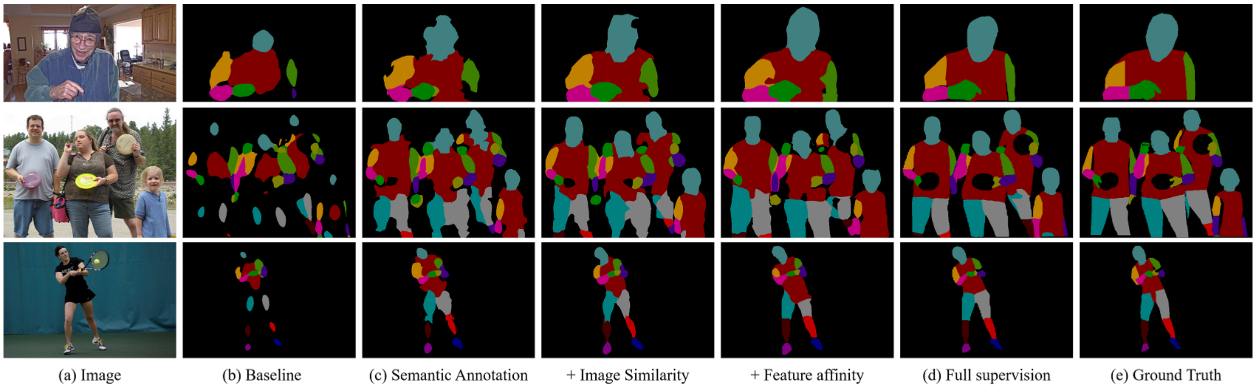


Figure 2.9: Our segmentation results get better with more types of regularizations. We compare visual results by adding more regularizations. As we introduce more relationships for regularization, we observe significant improvement and our results are visually closer to fully supervised counterparts.

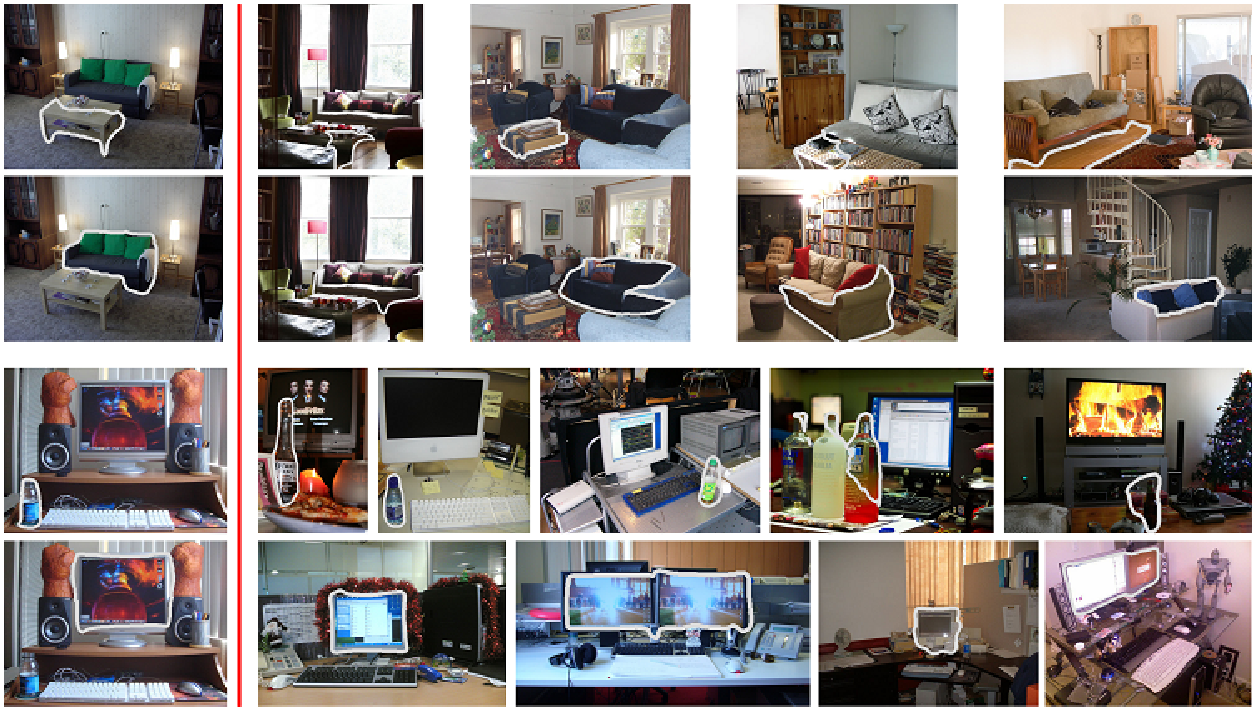


Figure 2.10: Visual examples of nearest neighbor segment retrievals. We observe that retrieved segments (right) appear in the similar semantic context as the query segments (left). For examples, given a bottle next to a desktop, our model retrieves bottles also next to a desktop.

Chapter 3

Unsupervised Hierarchical Semantic Segmentation with Multiview Cosegmentation and Clustering Transformers

3.1 Introduction

Semantic segmentation requires figuring out the semantic category for each pixel in an image. Learning such a segmenter from unlabeled data is particularly challenging, as neither pixel groupings nor semantic categories are known.

If pixel groupings are known, semantic segmentation is reduced to an unsupervised image (segment) recognition problem, to which contrast learning methods [74, 86, 87, 88] could apply, on computed segments instead of images.

If semantic categories are known, semantic segmentation is reduced to a weakly supervised segmentation problem with coarse annotations of image-level tags; pixel labeling can be predicted from image classifiers [38, 30].

The fundamental task of unsupervised semantic segmentation is *grouping*, not *semantics* in terms of *naming*, which is unimportant other than the convenience of tagging segments in the same or different groups. The challenge of unsupervised semantic segmentation is to discover groupings within and across images that capture object- and view-invariance of a category without external supervision, so that (Fig. 3.1): **1)** A baby's face and body are parts of a whole in the same image; **2)** The whole baby is separated from the rest of the image; **3)** A baby instance is more similar to another baby instance than to a cat instance, despite their different poses, illuminations, and backgrounds.

Several representative approaches have been proposed for tackling this challenge under different assumptions.

- **Visual similarity:** SegSort [48] first partitions each image into segments based on contour

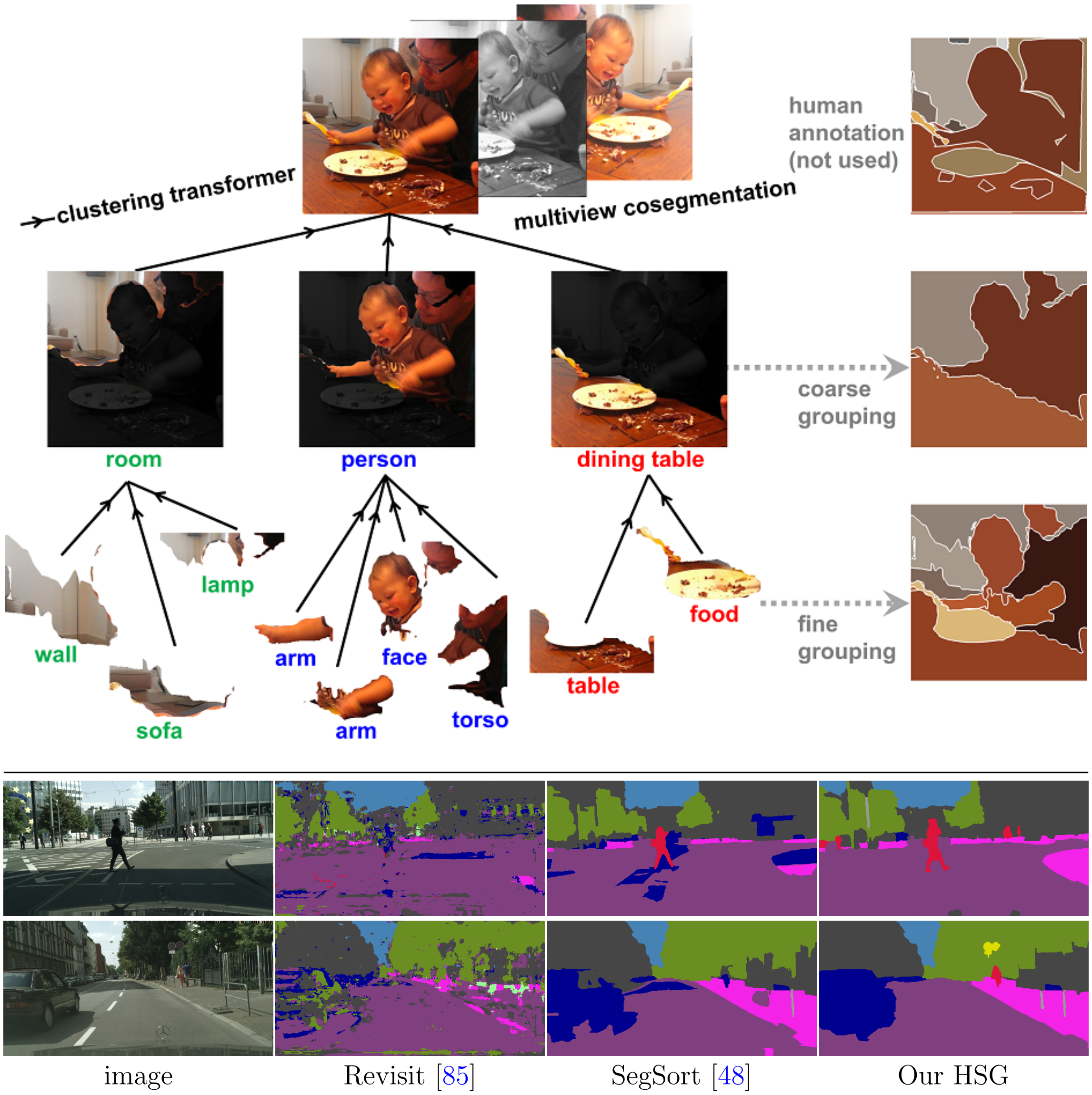


image Revisit [85] SegSort [48] Our HSG

Figure 3.1: We develop an unsupervised semantic segmentation method by embracing the ambiguity of grouping granularity and desiring hierarchical grouping consistency for unsupervised segmentation. **Top:** We formulate it as a pixel-wise feature learning problem, such that a good feature must be able to best reveal any level of grouping in a consistent and predictable manner. We bootstrap feature learning from multiview cosegmentation and enforce grouping consistency with clustering transformers. **Bottom:** Our method can not only deliver *hierarchical* semantic segmentation, but also outperform the state-of-the-art unsupervised segmentation methods by a large margin. Shown are sample Cityscapes results.

cues and then by segment-wise contrastive learning discovers clusters of visually similar segments. However, semantics by visual similarity is far too restrictive: A semantic whole is often made up of visually dissimilar parts. Parts of *body* such as *head* and *torso* look very different; it is not their visual similarity but their spatial adjacency and statistical co-occurrence that bind them together.

- **Spatial stability:** IIC [89] maximizes the mutual information between clusterings from two views of the same image related by a known spatial transformation, enforcing stable clustering while assuming that a fixed number of clusters are equally likely within an image. It works best for coarse and balanced texture segmentation and has major trouble scaling up with the scene complexity.
- **Image-wise feature learning:** [90, 85] train representations on object-centric datasets with multiscale cropping to sharpen the representation within the image. These methods do not work well on scene-centric datasets where an image has more than one dominant semantic class.

Grouping as well as semantics naturally have different levels of granularity: A *hand* is an articulated configuration of a *palm* and five *fingers*, likewise a *person* of a *head*, a *torso*, two *arms*, and two *legs*. Such an inherent grouping hierarchy poses a major challenge: Which level should an unsupervised segmentation method target at and what is the basis for such a determination? Existing methods avoid this ambiguity and treat it as either a factor outside the segmentation modeling, or an aspect of secondary concern.

Our key insight is that the inherent hierarchical organization of visual scenes is not a nuisance for scene parsing, but a universal property that we can exploit and desire for unsupervised segmentation. This idea has previously led to a general image segmenter that handles texture and illusory contours through edges entirely without any explicit characterization of texture or curvilinearity [91]. We now advance the concept to data-driven representation learning: A good representation shall reveal not just a particular level of grouping, but any level of grouping in a consistent and predictable manner across different levels of granularity.

We approach unsupervised semantic segmentation as an unsupervised pixel-wise feature learning problem. Our objective is to best produce a consistent hierarchical segmentation for each image in the entire dataset based entirely on hierarchical clusterings in the feature space (Fig. 3.1). Specifically, given the pixel-wise feature, we perform hierarchical groupings *within* and *across* images and their transformed versions (i.e.,*views*). In turn, groupings at each level impose a desire on how the feature should be improved to maximize the discrimination among different groups.

Our model has two novel technical components: **1) Multiview cosegmentation** is to not only enforce spatial consistency between segmentations across views, but also bootstrap feature learning from visual similarity and co-occurrences in a simpler clean setting; **2) Clustering transformers** are used to enforce semantic consistency across different levels of the feature grouping hierarchy.

To summarize, our work makes three contributions.

1. **We deliver the first unsupervised hierarchical semantic segmentation** method that can produce parts and wholes in a data-driven manner from an arbitrary collection of images, whether they come from object-centric or scene-centric datasets.
2. **We are the first to embrace the ambiguity of grouping granularity** and exploit the inherent grouping hierarchy of visual scenes to learn a pixel-wise feature representation for unsupervised segmentation. It can thus discover semantics based on not only visual similarity but also statistical co-occurrences.
3. **We outperform existing unsupervised (hierarchical) semantic segmentation methods by a large margin** on not only object-centric but also scene-centric datasets.

3.2 Related Work

Image segmentation refers to the task of partitioning an image into visually coherent regions. Traditional approaches often consist of two steps: extracting local features and clustering them based on different criteria, *e.g.,* , mode-finding [92, 78], or graph partitioning [93, 94, 95, 96, 97].

Hierarchical image segmentation has been supervisedly learned from how humans perceive the organization of an image [77]: While each individual segmentation targets a particular level of grouping, the collection of individual segmentations present the perceptual hierarchy statistically.

A typical choice for representing a hierarchical segmentation is contours: They are first detected to sharply localize region boundaries [98, 76] and can then be removed one by one to reveal coarser segmentations (OWT-UCM [77]).

Such models are trained on individual ground-truth segmentations, hoping that coarse and fine-grained organization would emerge automatically from common and rare contour occurrences respectively in the training data.

In contrast, our model is trained on multi-level segmentations unsupervisedly discovered by feature clustering, and it also operates directly on segments instead of contours.

Semantic segmentation refers to the task of partitioning an image into regions of different semantic classes. Most deep learning models treat segmentation as a spatial extension of image recognition and formulate it as a pixel-wise classification problem. They are often based on Fully Convolutional Networks [99, 100, 80], incorporating information from multiple scales [101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 30, 111].

SegSort [48] does not formulate segmentation as pixel-wise labeling, but pixel-segment contrastive learning that operates directly on segments delineated by contours. It learns pixel-wise features in a non-parametric way, *with* or *without* segmentation supervision. SPML [30] extends it to unify segmentation with various forms of weak supervision: image-level tags, bounding boxes, scribbles, or points.

Unsupervised semantic segmentation has been modeled by non-parametric methods using statistical features and graphical models [71, 72, 73]. For example, [71] proposes to discover region boundaries by mining the statistical differences of matched patches in coarsely aligned images.

There are roughly three lines of recent unsupervised semantic segmentation methods. **1)** One way is to increase the location sensitivity of the feature learned from images [74, 86, 87, 88], by either adding an additional contrastive loss between pixels based on feature correspondences across views [90], or using stronger augmentation and constrained cropping [85, 112]. **2)** A pixel-level *feature* encoder can be learned directly by maximizing discrimination between pixels based on either contour-induced segments [48] or region hierarchies [113] derived from OWT-UCM [77]. Segmentation is indicated by pixel feature similarity and semantic labels can be inferred from retrieved nearest neighbours in a labeled set. **3)** A pixel-wise *cluster* predictor can be directly learned by maximizing the mutual information [114, 115] between cluster predictions on augmented views of the same instance at corresponding pixels [89, 116].

Our model advances pixel-wise feature learning methods [48, 30, 117]: It contrasts features based on feature-induced hierarchical groupings themselves, and most strikingly, directly outputs consistent hierarchical segmentations.

3.3 Hierarchical Segment Grouping (HSG)

We approach unsupervised semantic segmentation as an unsupervised pixel-wise feature learning problem (Fig. 3.2). The basic idea is that, once every pixel is transformed into a point in the feature space, image segmentation becomes a point clustering problem.

Semantic segmentation and feature clustering form a pair of dual processes: **1)** Clustering of feature X defines segmentation G in each image: Pixels with features in the same (different) clusters belong to the same (different) semantic regions. This idea is used to co-segment similar images given handcrafted features [118, 119, 120]. **2)** Segmentation G defines the similarity of feature X: A pixel should be mapped close to its own segment group and far from other segment groups in the feature space. This idea is used to learn the pairwise feature similarity [121] and pixel-wise feature [48, 30] given segmentations.

Our key insight is that a good representation shall reveal not just a particular level of grouping – as past co-segmentation methods have explored, but any level of grouping in a consistent and predictable manner. If we embrace the ambiguity of grouping granularity that all previous methods have avoided and desire the consistency of hierarchical semantic segmentation on the pixel-wise feature, we address not only the shortcoming of cosegmentation, but also provide a joint feature-segmentation learning solution.

Specifically, while there is no supervision available for either feature X or segmentation G, we can desire that: **1)** each segmentation separates features well and **2)** the coarser segmentation defined by next-level feature clusters simply *merges* the current finer segmentation. These strong constraints guide the feature learning towards quality hierarchical segmentations, thereby better capturing semantics.

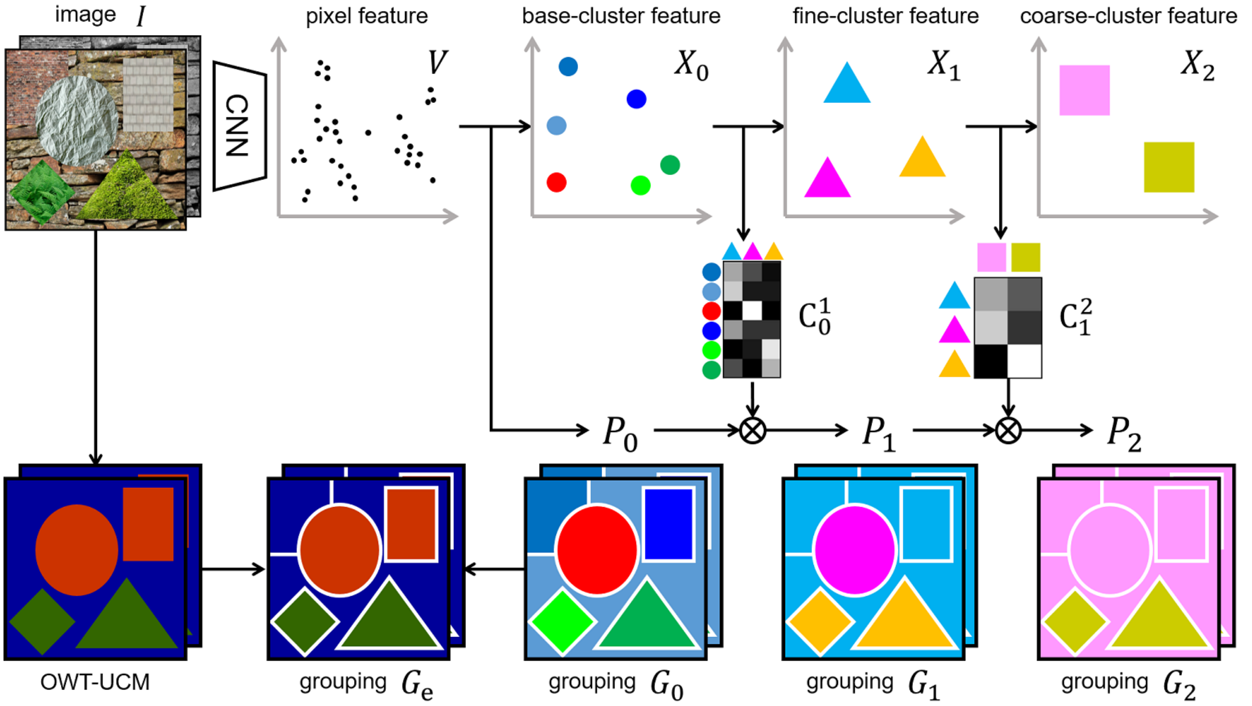


Figure 3.2: **Method overview**. We aim to learn a CNN that maps each pixel to a point in the feature space V such that successively derived cluster features X_0, X_1, X_2 produce good and consistent hierarchical pixel groupings G_e, G_1, G_2. Their consistency is enforced through clustering transformers C_l^{l+1}, which dictates how feature clusters at level l map to feature clusters at level $l+1$. Note that G_0 results from clusters of V, and G_e from OWT-UCM edges. P_l is the probabilistic version of G_l, and G_l the winner-take-all binary version of P_l; $P_0 \sim G_0$. For $l \geq 0$, P_{l+1} results from propagating P_l by C_l^{l+1}. Groupings G_e, G_1, G_2 in turn impose desired feature similarity and drive feature learning. We co-segment multiple views of the same image to capture spatial consistency, visual similarity, statistical co-occurences, and semantic hierarchies.

Our model has two components: **1)** multiview cosegmention to robustify feature clustering against spatial transformation and appearance variations of visual scenes, and **2)** clustering transformers to enforce consistent semantic segmentations across different levels of the feature grouping hierarchy. Both are necessary for mapping pixel features to segmentations, which in turn impose desired pairwise attraction and repulsion on the pixel features.

In the following, we first introduce our contrastive feature learning loss given any grouping G, and then describe how we obtain three kinds of groupings within and across images, and how we evaluate their goodness of grouping and enforce their consistency.

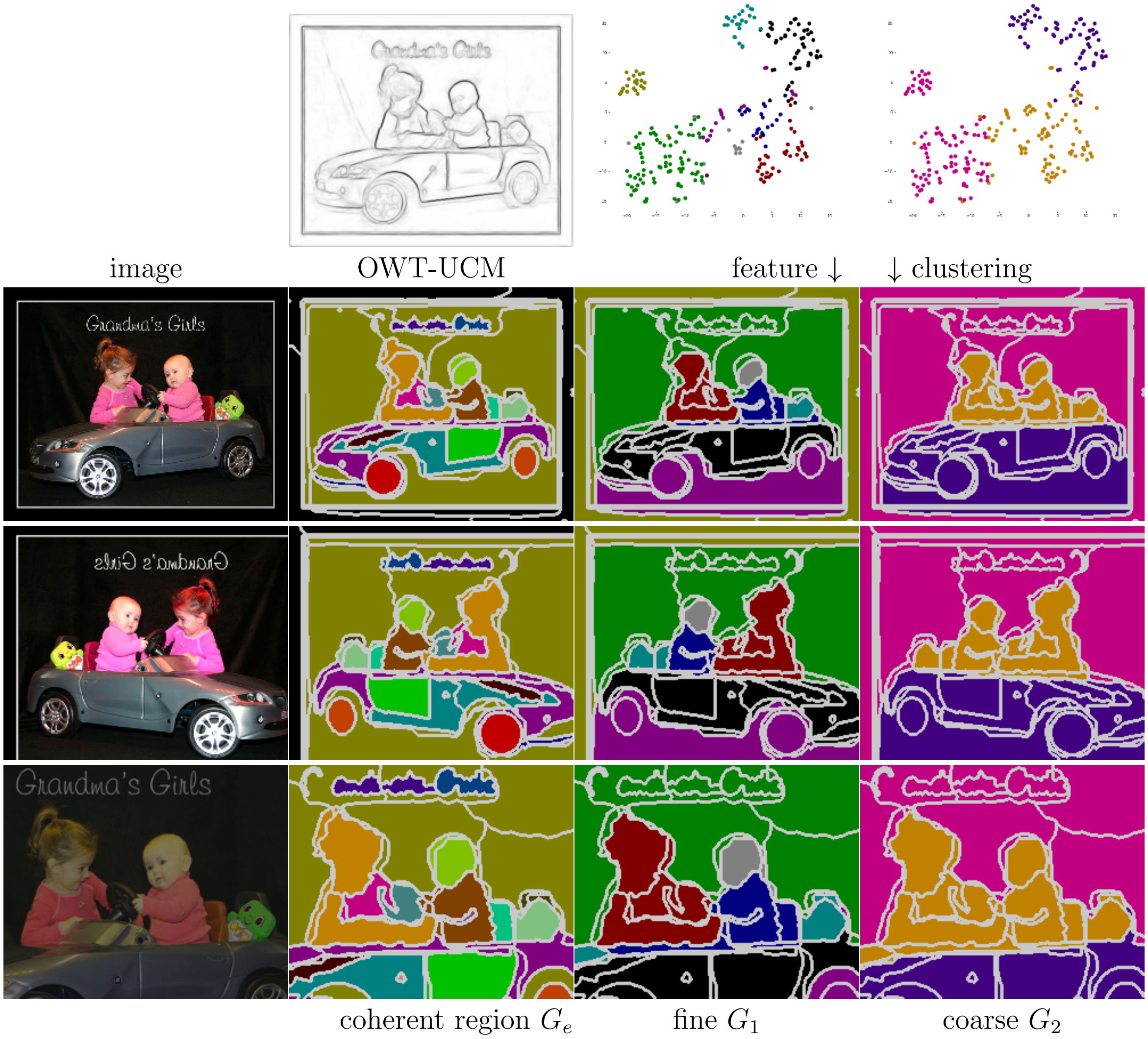


Figure 3.3: We co-segment multiple views (Column 1) of the same image by OWT-UCM edges (G_e, Column 2) or by feature clustering at fine and coarse levels (G_1, G_2, Columns 3-4). White lines mark the segments derived from pixel feature clustering and OWT-UCM edges. The color of feature points (pixels) mark grouping in the feature space (segmentation in the image) consistently across rows in the same column, per spatial transformations between views. G_2's coarse segmentations simply merge G_1's fine segmentations, their consistency enforced by our clustering transformers. Minimizing $\mathcal{L}_f(G_e), \mathcal{L}_f(G_1), \mathcal{L}_f(G_2)$ ensures respectively that our learned feature is grounded in low-level coherence, yet with view invariance, and capable of capturing semantics at multiple levels and producing hierarchical segmentations.

Pixel-Segment Contrastive Feature Learning

We learn a pixel-wise feature extraction function f as a convolutional neural network (CNN) with parameters θ. It transforms image I to its pixel-wise feature V. Let $\boldsymbol{v}_i$ be the *unit-length* feature vector at pixel i of image I:

$$\boldsymbol{v}_i = f_i(I;\theta), \quad \|\boldsymbol{v}_i\| = 1. \tag{3.1}$$

Suppose that I is partitioned into segments (Fig. 3.3). Let $\boldsymbol{u}_s$ be the feature vector for segment s, defined as the (length-normalized) average pixel feature within the segment:

$$\boldsymbol{u}_s \propto \text{mean}\left(\boldsymbol{v}_i : i \text{ in segment } s\right), \quad \|\boldsymbol{u}_s\| = 1 \tag{3.2}$$

Consider a batch of images and their pixel groupings $\{(I, G)\}$. We want to learn the right feature mapper f such that all the pixels form distinctive clusters in the feature space, each corresponding to a different semantic group.

We follow [48, 30] to formulate desired feature-wise attraction and repulsion *not between pixels*, but *between pixels and segments*. Such contrastive learning across granularity levels reduces computation, improves balance between attraction and repulsion, and is more effective [88].

Our contrastive feature learning loss to minimize is:

$$\mathcal{L}_f(G) = \sum_i -\log \frac{\sum\limits_{s\in G_i^+} \exp \frac{\boldsymbol{v}_i^\top \boldsymbol{u}_s}{T}}{\sum\limits_{s\in G_i^+} \exp \frac{\boldsymbol{v}_i^\top \boldsymbol{u}_s}{T} + \sum\limits_{s\in G_i^-} \exp \frac{\boldsymbol{v}_i^\top \boldsymbol{u}_s}{T}} \tag{3.3}$$

where T is a temperature hyper-parameter that controls the concentration level of the feature distribution. Ideally, $\boldsymbol{v}_i$ should be attracted to segments in the positive set G_i^+ and repelled by segments in the negative set G_i^-.

Our batch of images consists of several augmented *views* of some training instances. For pixel i in a particular view of image I, G_i^+ includes segments of the same semantic group in any view of image I except i's own segment, in order to achieve within-instance invariance, whereas G_i^- includes segments of different semantic groups in any view of I, *and* segments of training instances other than I, in order to maximize between-instance discrimination [74, 48].

Consistent Segmentations by View & Hierarchy

From pixel feature V, we compute feature grouping G_0 and cluster feature X_0. Our initial pixel grouping G_e is based on OWT-UCM edges detected in the image. Next-level cluster feature X_{l+1} and grouping G_{l+1} are predicted from G_l with ensured consistency. We use three levels for the sake of illustration (Fig. 3.3), but our procedure can be repeated for more (coarser) levels.

Base cluster feature X_0 and grouping G_0, G_e. We segment each view of I by clustering pixel features, resulting in base grouping G_0 and cluster (centroid) feature X_0 (Fig. 3.2).

During training but *not* testing, we segment image I into a fixed number of coherent regions according to its OWT-UCM edges [122], based on which we split each G_0 region to obtain edge-conforming *segments* [48] marked by white lines in Fig. 3.3. For training, we obtain pixel grouping G_e by inferring the coherent region segmentation according to how each view is spatially transformed from I.

Minimizing $\mathcal{L}_f(G_e)$ encourages the feature to be similar not only for different pixels of similar appearances in the image, but also for corresponding pixels of different appearances across views of I. The former grounds the feature f at respecting low-level appearance coherence, whereas the latter develops view invariance in the feature.

Next-level cluster feature X_{l+1} and grouping G_{l+1}. Now we have grouping G_0 in the feature space of V, and for each cluster, we obtain its centroid feature in X_0. We model how cluster feature X_l maps to cluster feature X_{l+1}, which corresponds to how segmentation at level l maps to segmentation at level $l+1$ in the image.

We adopt a probabilistic framework, where any feature point $\boldsymbol{x}$ has a (soft assignment) probability belonging to a group determined by its cluster centroid. Let $P_l(a)$ be the probability of $\boldsymbol{x}$ in group a at level l:

$$P_l(a) = \text{Prob}(G_l = a \,|\, \boldsymbol{x}). \tag{3.4}$$

To ensure that feature points in the same group remain together at the next level, we introduce group transition probability $C_l^{l+1}(a, b)$, the transition probability from group a at level l to group b at level l+1:

$$C_l^{l+1}(a, b) = \text{Prob}(G_{l+1} = b \,|\, G_l = a). \tag{3.5}$$

Per the Bayesian rule, we have:

$$P_{l+1}(b) = \sum_a P_l(a) \cdot C_l^{l+1}(a, b). \tag{3.6}$$

Writing P_l as a row vector, we can derive the soft group assignment P_{l+1} for cluster feature X_0 at level l+1:

$$P_{l+1} = P_l \times C_l^{l+1} = P_0 \times C_0^1 \times C_1^2 \times \cdots \times C_l^{l+1}. \tag{3.7}$$

Clustering Transformers. C_l^{l+1} is defined on multiview cosegmentation of each instance. We learn a function, in terms of a transformer [123], to naturally capture feature group transitions for all the training instances. It enables more consistent grouping compared to non-parametric clustering methods such as KMeans, NCut [124], and FINCH [125].

Our clustering transformer from level l to l+1 maps group centroid feature X_l to the next-level group centroid feature X_{l+1}, and simultaneously outputs the group transition probability C_l^{l+1} (Fig. 3.4).

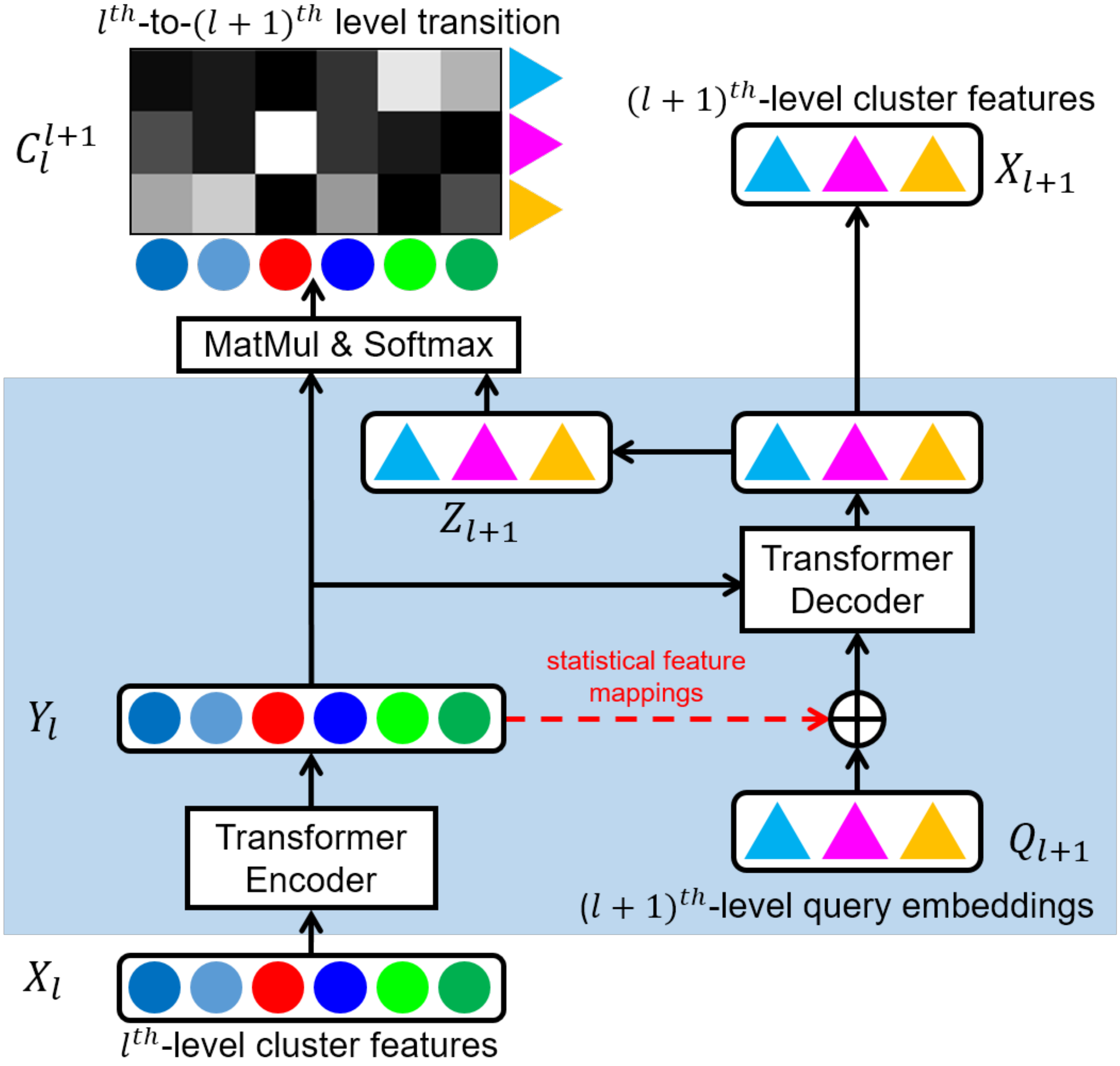

l^{th}-to-$(l+1)^{th}$ level transition
C_l^{l+1}
$(l+1)^{th}$-level cluster features
X_{l+1}
MatMul & Softmax
Z_{l+1}
Transformer Decoder
statistical feature mappings
Y_l
Transformer Encoder
Q_{l+1}
$(l+1)^{th}$-level query embeddings
X_l
l^{th}-level cluster features

Figure 3.4: Our clustering transformer enforces grouping consistency across levels by mapping feature X_l to X_{l+1} with feature transition C_l^{l+1}. X_{l+1} and C_l^{l+1} are learned simultaneously. Shown here for level $l=0$ in Fig. 3.2, the transformer encoder takes inputs X_l and outputs contextualized feature Y_l. The transformer decoder takes learnable inputs from query embeddings Q_{l+1}, and outputs X_{l+1} and additionally projected feature Z_{l+1}. The transition is predicted as: $C_l^{l+1} = \text{softmax}\left(\frac{1}{\sqrt{m}} Y_l^\top Z_{l+1}\right)$; m is the feature dimension. Statistical feature mapping: Calculate Y_l's mean and std, transform them by fc layers, and add to Q_{l+1} for instance adaptation.

Consistent feature groupings. At level $l=0$, P_0 has binary values, indicating hard grouping G_0. For next level l, we compute P_{l+1} by propagating P_l with our clustering transformer C_l^{l+1}, which also outputs X_{l+1}. We obtain G_{l+1} by binarizing P_{l+1} with winner-take-all. By decreasing the number of groups as l increases, we obtain consistent fine to coarse segmentations G_1, G_2 (Fig. 3.2).

Minimizing $\mathcal{L}_f(G_1)$ and $\mathcal{L}_f(G_2)$ encourages the feature f to capture semantics at multiple levels and produce consistent hierarchical segmentations (Fig. 3.3).

Goodness of Grouping

While clustering transformers ensure grouping consistency across levels, we still need to drive feature learning towards good segmentations. We follow [126] and supervise our transformer with modularity maximization [127] and collapse regularization. The former seeks a partition that results higher (lower) in-cluster (out-cluster) similarity than the total expectation, whereas the latter encourages partitions of equal sizes. We additionally maximize the separation between cluster centroids.

We first build a sparsified graph based on pairwise feature similarity for X_0. Let e be the number of edges in this graph, n_l the number of centroids in X_l, A the $n_0 \times n_0$ connection matrix for edges, D the $n_0 \times 1$ degree vector of A, M_l the $n_0 \times n_l$ soft assignment matrix where each row is P_l for a centroid of X_0, and $\boldsymbol{z}_{l,k}$ the normalized k-th feature of Z_l in Fig. 3.4. Our goodness of grouping loss is:

$$\mathcal{L}_g = \sum_{l\geq 1} \underbrace{\frac{-1}{2e}\operatorname{trace}(M_l^\top (A - \frac{1}{2e}DD^\top) M_l)}_{\text{maximize modularity}} + \underbrace{\frac{\sqrt{n_l}}{n_0}\|1^\top M_l\|_F - 1}_{\text{collapse regularization}}$$
$$+ \underbrace{\frac{1}{n_l}\sum_k -\log \frac{\exp(\boldsymbol{z}_{l,k}^\top \boldsymbol{z}_{l,k})}{\sum_j \exp(\boldsymbol{z}_{l,k}^\top \boldsymbol{z}_{l,j})}}_{\text{maximize centroid separation}} \tag{3.8}$$

Model Overview: Training and Testing

Our model (Fig. 3.5) is trained with the contrastive feature learning losses given edge-based grouping G_e and multi-level feature-based grouping G_l, and the goodness of grouping loss, weighted by λ_E, λ_F, and λ_G respectively:

$$\mathcal{L}(f) = \lambda_E \mathcal{L}_f(G_e) + \lambda_F \sum_{l\geq 1} \mathcal{L}_f(G_l) + \lambda_G L_g. \tag{3.9}$$

For testing, the same pipeline with the pixel feature CNN and clustering transformers predicts hierarchical segmentations $\{G_l\}$. To benchmark segmentation performance given a labeled set, We follow [48] and predict the labels using k-nearest neighbor search for each segment feature.

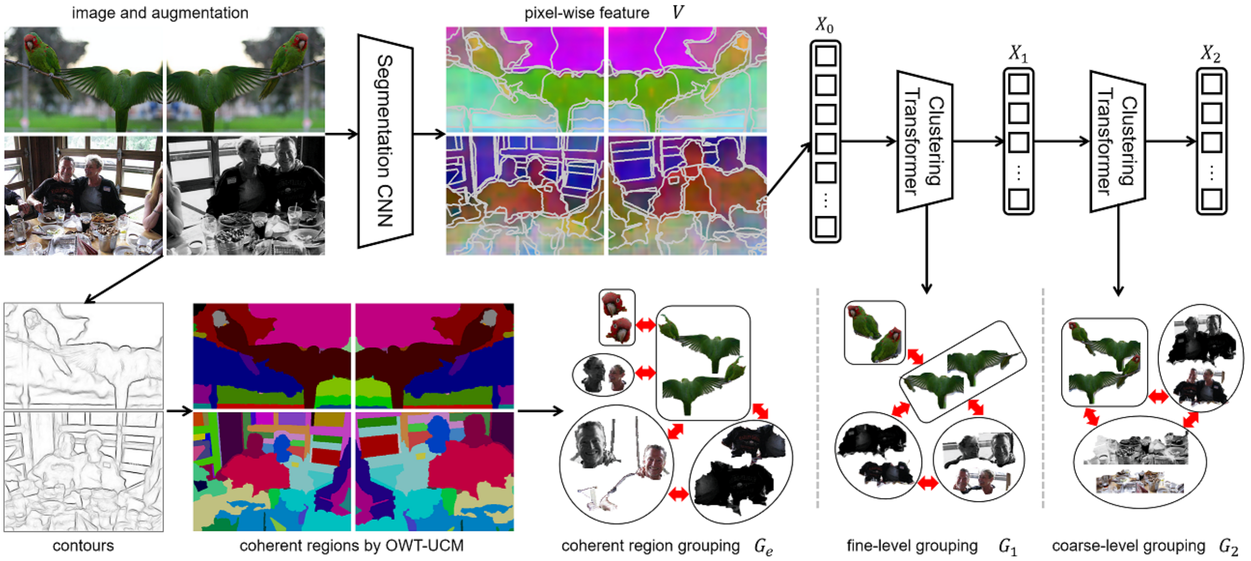


Figure 3.5: Our model consists of two essential components: **1)** multiview cosegmentation and **2)** hierarchical grouping. We first produces pixel-wise feature V, from which we cluster to get base cluster feature X_0 and grouping G_0. Each G_0 region is split w.r.t coherent regions derived by OWT-UCM procedure, which is marked by the white lines. We create three groupings–G_e, G_1 and G_2 in multiview cosegmentation fashion. We obtain G_e by inferring the coherent region segmentation according to how each view is spatially transformed from the original image. Starting with input X_0 of an image and its augmented views, we conduct feature clustering to merge G_0 into G_1, and then, G_1 into G_2. Based on G_e, G_1 and G_2, we formulate a pixel-to-segment contrastive loss for each grouping. Our HSG learns to generate discriminative representations and consistent hierarchical segmentations for the input images.

3.4 Experiments

We benchmark our model on two tasks: unsupervised semantic segmentation and hierarchical image segmentation, the first on five major object- and scene-centric datasets and the second on Pascal VOC. We conduct ablation study to understand the contributions of our model components.

We adopt FCN-ResNet50 as the common backbone architecture. The FCN head consists of 1×1 convolution, BatchNorm, ReLU, and 1×1 convolution. Specifically, we follow DeepLabv3 [101] to set up the dilation and strides in ResNet50. We set Multi_Grid to $(1, 2, 4)$ in res5. The output_stride is set to 16 and 8 during training and testing. We do not use any pre-trained models, but train our models from scratch on each dataset. Ground-truth annotations are not for training but only for testing and evaluation's sake.

Datasets

Pascal VOC 2012 [15] is a generic semantic segmentation dataset of 20 object category and a background class. It consists of $1,464$ and $1,449$ images for training and validation. We follow [80] to augment the training data with additional annotations [128], resulting in $10,582$ training images. Following [85], we do not train but only inference on VOC.
MSCOCO [16] is a complex scene parsing dataset with 80 object categories. Objects are embedded in more complex scenes, with more objects per image than Pascal (7.3 vs. 2.3). Following [90, 85], we use *train2017* split ($118,287$ images) for training and test on the VOC validation set.
Cityscapes [27] is an urban street scene parsing dataset, with 19 stuff and object categories. Unlike MSCOCO and VOC where classes are split by scene context, Cityscapes contains similar street scenes covering almost all 19 categories. The train/test split is $2,975/500$.
KITTI-STEP [129] is a video dataset for urban scene understanding, instance detection and object tracking. It has pixel-wise labels of the same 19 categories as Cityscapes. There are 12 and 9 video sequences for training and validation, or $5,027$ and $2,981$ frames.
COCO-stuff [130] is a scene texture segmentation dataset, a subset of MSCOCO. As [89, 116], we use 15 coarse *stuff* categories and reduce the dataset to 52K images with at least 75% stuff pixels. The train/test split is $49,629/2,175$.
Potsdam [131] is a dataset for aerial scene parsing. The raw 6000×6000 image is divided into 8550 RGBIR 200×200 patches. There are 6 categories (*roads, cars, vegetation, trees, buildings, clutter*). The train/test split is $7,695/855$.

Hierarchical Clustering Transformer Architecture

We present more details about the model architecture of our clustering transformer. We mostly follow [123] to implement the transformer. The detailed architecture of the clustering transformer is presented in Fig. 3.4. The $(l+1)^{th}$-level transformer takes X_l as inputs and forwards to the encoder. The encoder contextually updates X_l to Y_l based on the pairwise correlation information of X_l. Meanwhile, the decoder takes a set of query embeddings Q_{l+1} as inputs and outputs the next-level cluster centroids. Q_{l+1} can be considered as the initial representations of next-level clusters. As the clusterings should adapt with input statistics, we calculate the 'mean' and 'std' of Y_l, followed by fc layers, and sum them with Q_{l+1} before inputting to the decoder. The decoder contextually updates Q_{l+1} to the next-level cluster centroids X_{l+1}, which become the inputs to the next-level transformer. To calculate the clustering assignment, we do not use X_{l+1} but Z_{l+1}, which shares the decoder layers with X_{l+1} but transformed by a separate fc layer. The soft clustering assignments are calculated as: $C_l^{l+1} = \text{softmax}(\frac{1}{\sqrt{m}} Y_l^\top Z_{l+1})$; m is the feature dimension.

In particular, we replace LayerNorm with BatchNorm. We set number of heads to 4 in the attention module, and use 2 encoder (decoder) layers in each encoder (decoder) module. We set drop_out rate to 0.1 during training. For query embeddings Q_l at level l, we randomly initiate and update them thru SGD.

In the clustering loss, the affinity matrix A among base level feature X_0 is required to compute the modularity maximization loss. We construct A as a k-nearest neighbor (sparse) graph using the similarity of X_0, where the entry value is set to 1. A is a binarized affinity matrix of a sparsified graph. For MSCOCO/VOC/COCO-stuff/Potsdam, we set k to 2 within an image and its augmented views, respectively. In such a manner, we encourage segment groupings across views. On Cityscapes/KITTI-STEP, cropped patches from each image instance are less likely to overlap. Without enforcing groupings across views, we search top 4 nearest neighbors among views ($k = 4$).

Dataset	B.S	C.S	L.R	W.D	Epochs	λ_E	λ_G	λ_F
MSCOCO	128	224	0.1	0.0001	380	1.0	0.0	0.0
	48	448	0.008	0.0001	8	1.0	1.0	0.1
Cityscapes	32	448	0.1	0.0001	400	1.0	0.2	0.1
KITTI-STEP	48	448	0.1	0.0001	400	1.0	0.2	0.1
COCO-stuff	8	336	0.003	0.0005	5	1.0	0.2	0.1
Potsdam	8	200	0.003	0.0005	30	1.0	0.2	0.1

Table 3.1: Hyper-parameters for training on different datasets. Gray colored background indicates pre-training settings. B.S, C.S, L.R, W.D denote batch size, crop size, learning rate and weight decay.

Training, Testing and Inference

We first describe the same set of hyper-parameters shared across different datasets, and summarize the different settings in Table. 3.1.

For all the experiments, we set the dimension of output embeddings to 128, temperature T to $\frac{1}{16}$. We apply step-wise decay learning rate policy, with which learning rate is decayed by 32%, 56% and 75% of total training epochs. We obtain base-level grouping G_0 by iterating spherical KMeans algorithm over pixel-wise feature V for 15 steps and partition each cropped input to 4×4 segments. During training not testing, G_0 is then refined by coherent regions generated from the OWT-UCM procedure. For G_1 and G_2, we set n_1 and n_2 to 8 and 4. The whole framework is optimized using SGD. Notably, we only adopt rescaling, cropping, horizontal flipping, color jittering, gray-scale conversion, and Gaussian blurring for data augmentation. All the other different settings are presented in Table. 3.1. For fair comparison with corresponding baselines, we apply different settings for training on MSCOCO and COCO-stuff.

Particularly, for MSCOCO, we adopt a two-stage learning strategy. We first train the model with smaller crop size (224×224) and larger batch size (128), then fine-tune with larger crop size (448 × 448) and smaller batch size (48). The models are trained and fine-tuned for 380 and 8 epochs. We do not use spherical KMeans to generate image oversegmentation in the first stage of training.

For inference, we only use single-scale image. For unsupervised semantic segmentation, we follow [48] to conduct nearest neighbor search to predict the semantic segmentation. We apply spherical KMeans algorithm over V to derive pixel grouping G_0 and base cluster feature X_0. We search nearest neighbors using X_0 from the whole training dataset. We set n_0–the number of centroids in G_0, to 6×6, 12×24 and 6×12 on Pascal/COCO-stuff/Potsdam, Cityscapes and KITTI-STEP dataset. On Cityscapes and KITTI-STEP, we train the baselines with officially released code and test with our inference procedure. Otherwise, we report the numbers according to their papers.

We follow [48] and adopt the UCM-OWT procedure [77] to generate coherent region segmentations from contours. For MSCOCO and COCO-stuff, we follow [113] to detect edges by SE [122]. The detector is first pre-trained on BSDS dataset [18] with ground-truth edge labels. We start with threshold as 0.25 to binarize the UCM, followed by OWT-UCM to generate the segmentations. We gradually increase the threshold until the number of regions is smaller than 48. For Cityscapes/KITTI-STEP and Potsdam, we use PMI [132] to predict edges. The detector only considers co-occurring statistics among paired colors, and does not require any ground-truth label. The initial threshold is 0.05 and 0.5, which is increased until the number of regions is smaller than 1024 and 128.

Quantitative Results on Unsupervised Semantic Segmentation

All the models are trained from scratch and evaluated by IoU and pixel accuracy. For VOC, we follow baselines [85] to train on MSCOCO. Table. 3.2 shows that our method outperforms baselines by 6.8%, 7.9% and 2.5% in mIoU on VOC, Cityscapes, and KITTI-STEP validation sets respectively.

Note that methods relying on image-wise instance discrimination do not work well on Cityscapes and KITTI-STEP. Both datasets have urban street scenes with similar categories in each image. Our method can still discover semantics by discriminating regions among these images.

For texture segmentation on COCO-stuff and Potsdam, Tab. 3.3 shows that our method achieves huge gains, +26.8% and +18.1% over IIC [89] and AC [116] respectively.

Quantitative Results on Hierarchical Segmentation

We benchmark hierarchical segmentation with respect to ground-truth segmentation. We evaluate the overlapping of regions between predicted segmentations and ground truth within each image, known as *Segmentation Covering* [77]. However, such a metric scores performance with the number of pixels within each segment, and is thus easily biased towards large regions. For object-centric dataset VOC, a trivial all-foreground mask would rank high by the Covering metric.

Training set	MSCOCO		Cityscapes		KITTI-STEP	
Validation set	VOC		Cityscapes		KITTI-STEP	
Method	mIoU	Acc.	mIoU	Acc.	mIoU	Acc.
Moco [86]	28.1	-	15.3	69.5	13.7	60.3
DenseCL [90]	35.1	-	12.7	64.2	9.3	47.6
Revisit [85]	35.1	-	17.1	71.7	17.0	65.0
SegSort [48]	11.7	75.1	24.6	81.9	19.2	69.8
Our HSG	**41.9**	**85.7**	**32.5**	**86.0**	**21.7**	**73.8**

Table 3.2: Our method delivers better performance on different types of datasets. The results are reported on VOC, KITTI-STEP and Cityscapes val set, using IoU and pixel accuracy metrics. In VOC, object categories are separated according to image scenes. In Cityscapes and KITTI-STEP, images all come from urban street scene and thus contain mostly the same set of categories. Instance-discrimination methods apply image-wise contrastive loss, and learn less optimally on Cityscapes and KITTI-STEP, as image scenes are similar. Our HSG instead learns to discriminate regions at different scales and performs well on both types of datasets.

	COCO-stuff		Potsdam	
Method	mIoU	Acc.	mIoU	Acc.
DeepCluster 2018 [133]	-	19.9	-	29.2
Doersch 2015 [134]	-	23.1	-	37.2
Isola 2016 [135]	-	24.3	-	44.9
IIC [89]	-	27.7	-	45.4
AC [116]	-	30.8	-	49.3
SegSort [48]	16.4	49.9	35.0	59.0
Our HSG	**23.8**	**57.6**	**43.8**	**67.4**

Table 3.3: Our method outperforms baselines on both stuff region and aerial scene parsing datasets. The results are reported on COCO-stuff and Potsdam test set, using IoU and pixel accuracy metrics. We evaluate our model using nearest neighbor search. Our HSG achieves superior performance.

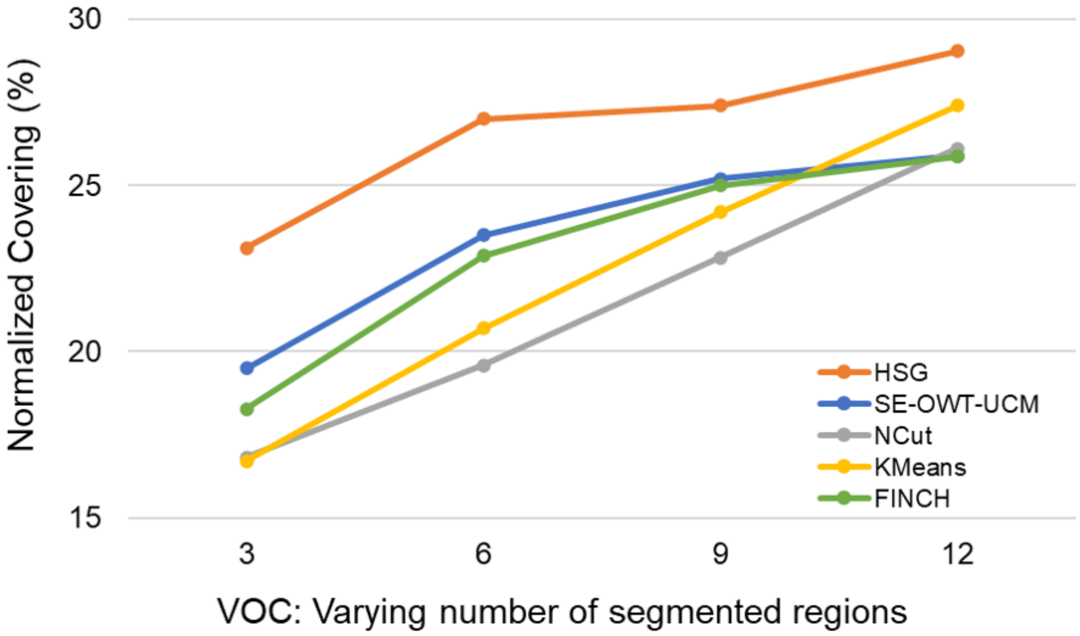


Figure 3.6: Our clustering transformers capture semantics at different levels of granularity. We compare to other clustering algorithms on VOC val set, using *Normalized Foreground Coverings* as metric. We exclude background regions for evaluation. Our HSG overlaps with ground truths more accurately.

We propose a *Normalized Foreground Covering* metric, by focusing on the foreground region and the overlap ratio instead of the overlap pixel count. To measure the average foreground region overlap ratio of a ground-truth segmentation S by a predicted segmentation S', we define:

$$\text{NFCovering}(S' \to S_{fg}) = \frac{1}{|S_{fg}|} \sum_{R \in S_{fg}} \max_{R' \in S'} \frac{|R \cap R'|}{|R \cup R'|} \tag{3.10}$$

where S_{fg} denotes the set of ground-truth foreground regions. Given a hierarchical segmentation, we report NFCovering at each level in the hierarchy. Our method outperform other clustering algorithms by large margin (see Fig. 3.6). Fig. 3.7 shows that our clustering transformers produce segmentations better aligned with the ground-truth foreground at every level.

Visual Results on Semantic Segmentation

Fig. 3.8 shows sample semantic segmentations on VOC (trained on MSCOCO), Cityscapes and KITTI-STEP. Compared to SegSort [48], our method retrieves same-category segments

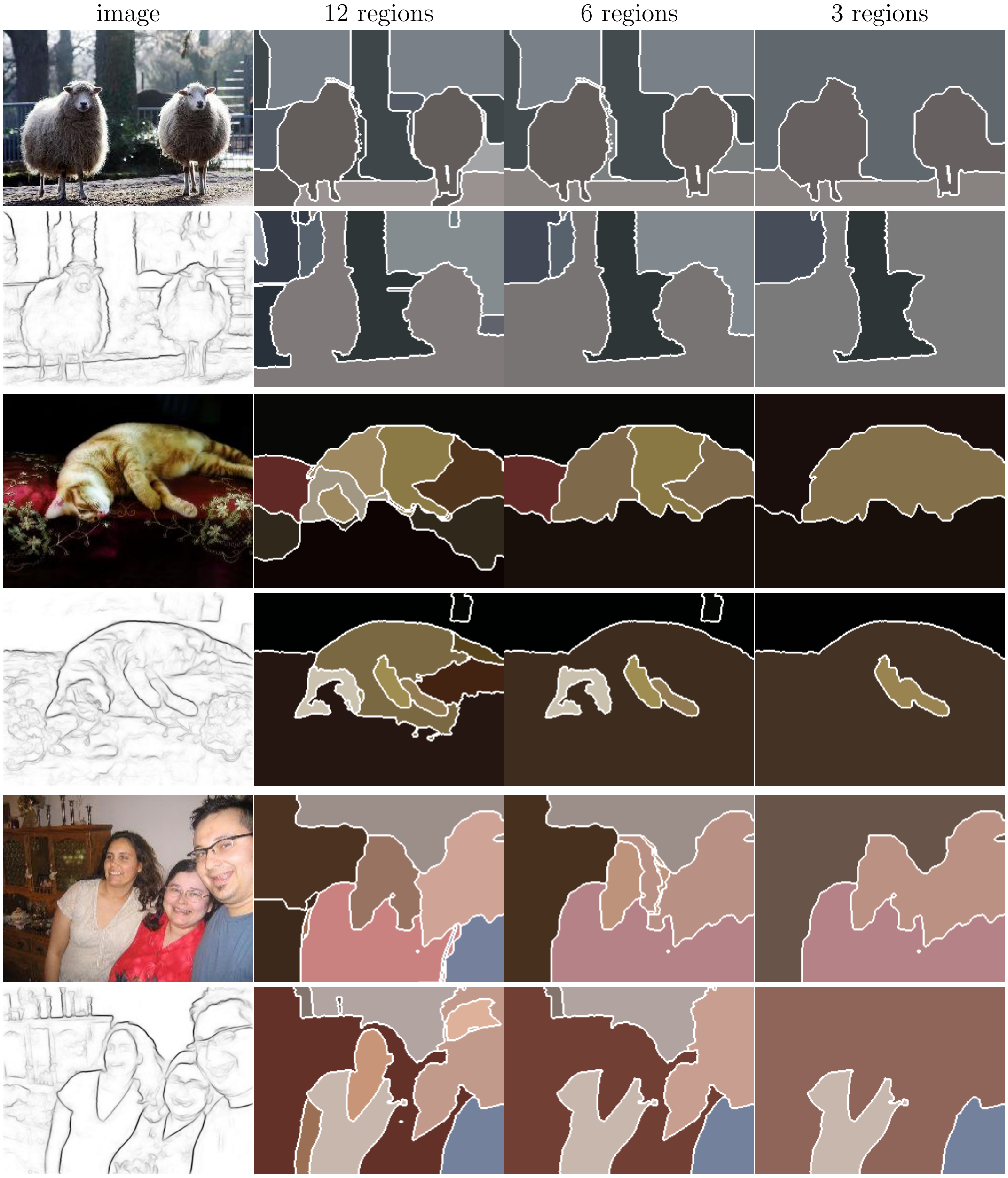


Figure 3.7: Our clustering transformers capture semantics at different levels of granularity. We present visual results to compare our hierarchical segmentation (top row) with SE [122]-OWT-UCM procedure (bottom row). We also show the detected edges at the leftmost figure in the bottom row. Each image is segmented into 12, 6, 3 regions. Our method reveals low-to-high level of semantics more consistently.

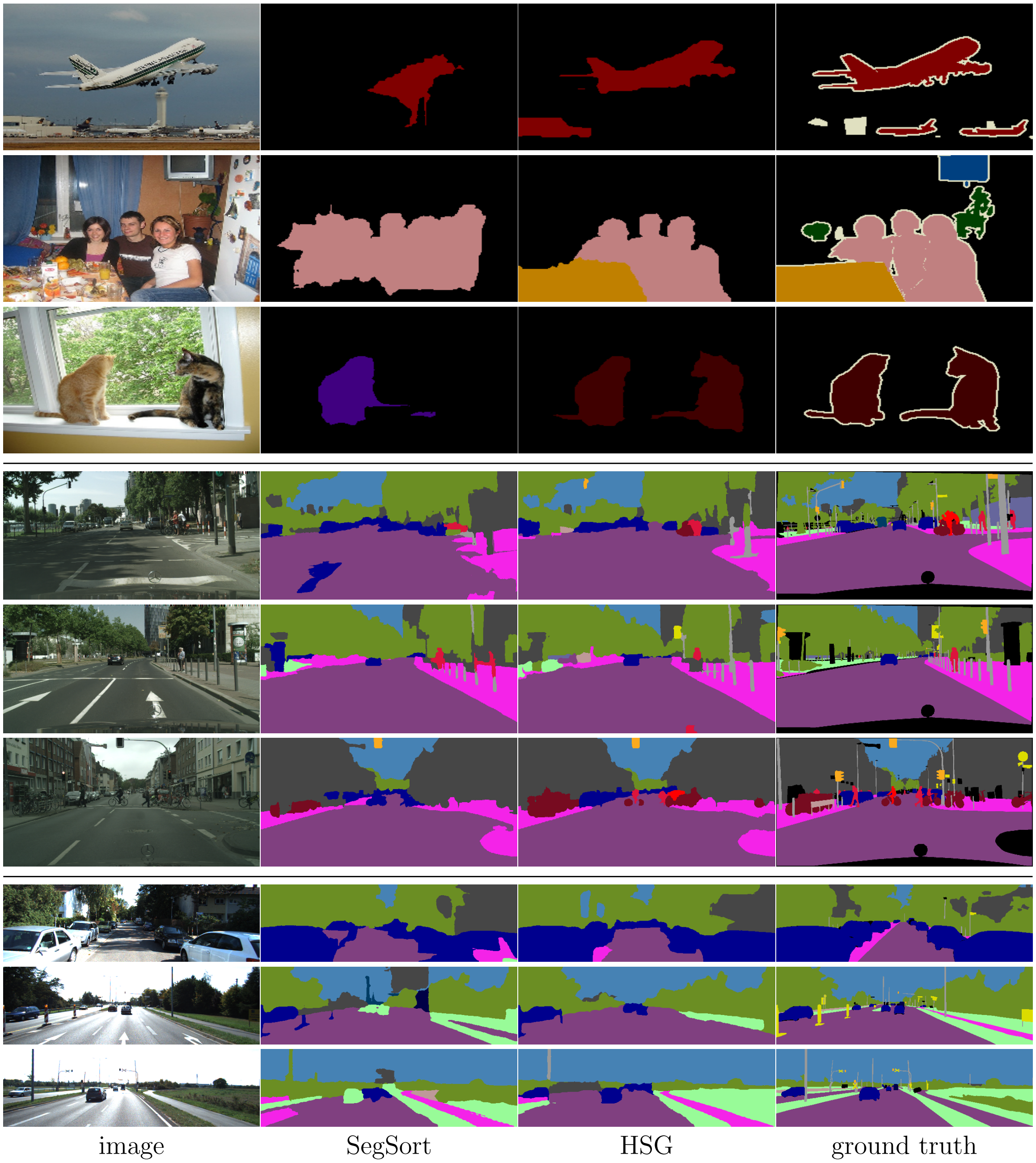


Figure 3.8: Our framework performs better on different types of datasets. From top to bottom every three rows are visual results from VOC, Cityscapes and KITTI-STEP dataset. The results are predicted via segment retrievals. Our pixel-wise features encode more precise semantic information than baselines.

more accurately. For larger objects or stuff categories, such as *airplane* or *road*, our results are more consistent within the region. Our segmentations are also better at respecting object boundaries.

Visual Results on Hierarchical Segmentation

We also compare our hierarchical segmentations with SE [122]-OWT-UCM, an alternative based entirely on low-level cues. Fig. 3.7 bottom shows that, when partitioning an image into 12, 6 and 3 regions, our segmentations follow the semantic hierarchy more closely.

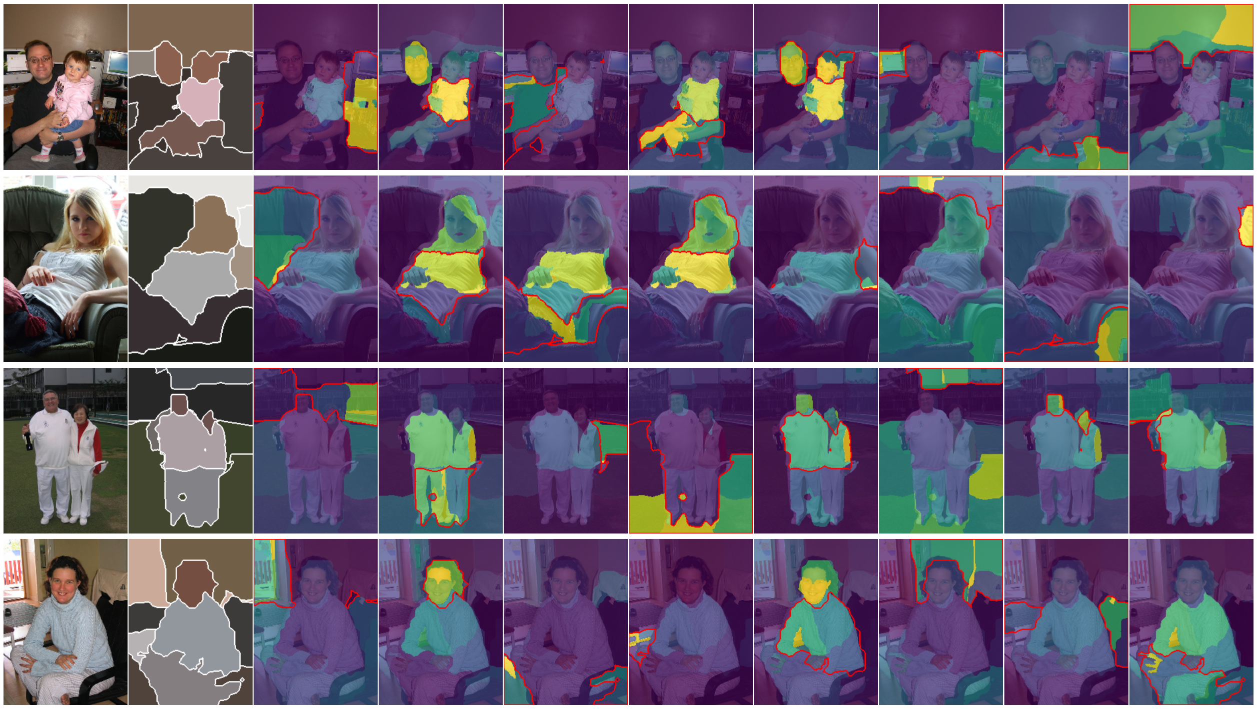


Figure 3.9: The multi-head attention maps reveal the fine-to-coarse semantic relationships among image segments. **From left to right:** input image, our feature-induced segmentation, attention maps in the decoder of our clustering transformers. We use a clustering transformer to partition each image into 8 clusters, and show the attention map (colored in *viridis* color maps) of each cluster to all the image segments. We observe these clusters correlate better with image segments that carry more similar semantic meanings, e.g., the 'head' cluster attends more to body parts than background regions. Such correlation information implies the next-level groupings: 'head' will be grouped with 'torso' instead of 'background'

Visual Results on Attention Maps from Decoder

We visualize the multi-head attention maps in the decoder of our clustering transformer. Such attention maps correspond to the correlation among cluster centroids and input segments. As shown in Fig. 3.9, we observe that each cluster attends to their cluster members, e.g. face and hair of the head region. Interestingly, we also see these clusters correlate better with image segments that carry more similar semantic meanings. For example, the 'head' cluster attends more to body parts than background regions. Such correlation information implies the next-level groupings: 'head' will be grouped with 'torso' instead of 'background'.

Visual Results on Contextual Retrievals

We reveal the encoding of visual context in our learned feature representations. We first conduct hierarchical segmentation using our clustering transformers to partition an image into fine and coarse regions. We then compute the unit-length average feature within each region and perform nearest neighbor search among the training dataset. Fig. 3.10 shows nearest neighbor retrievals at coarse (cyan) and fine (red) segmentations. The query and retrieved segments are generated at same level of partitioning. Strikingly, the feature representations at each level of grouping correlate with multiple levels of semantic meanings such as baseball players and their body parts.

We next demonstrate the contextual information of co-occurring objects encoded in our feature representations. We visualize the length-normalized average features of the 'person' category region on Pascal VOC 2012 dataset using tSNE [136]. We represent each 'person' feature with the co-occurring object categories, and observe that features in the similar semantic context are clustered. As shown in Fig. 3.11, we observe clusters of similar co-occurring object categories, such as a person riding a horse (in cerise) or a bike (in green), *etc.*

Ablation Study of Regularizations

Tab. 3.4 shows that our model improves consistently by adding the feature learning loss based on hierarchical groupings and the goodness of grouping loss. It also shows that multiview cosegmentation significantly improves the performance over a single image.

Ablation Study of Clustering Algorithms

Tab. 3.5 shows that our clustering transformers provide better regularization with hierarchical groupings than alternative non-parametric clustering methods.

Inference Latency on Clustering Transformer

We present the inference latency of different hierarchical clustering methods. We test on a 640 × 640 image, which is hierarchically partitioned into 25, 16, 9 and 4 segments. We

λ_E	λ_G	λ_F	single-view	multi-view
✓	-	-	13.0	40.9
✓	✓	-	13.8	41.7
✓	✓	✓	14.0	41.9

Table 3.4: Regularizing with our goodness of grouping loss and pixel-to-segment contrastive losses improves learned features. The results are reported over VOC val set, using IoU metric. Our resulted pixel features encode better semantic information.

Method	KMeans	NCut [124]	FINCH [125]	Our Transfomer
mIoU	41.2	41.3	40.6	**41.9**

Table 3.5: Our hierarchical clustering transformer follows semantics closer than other non-parametric clustering algorithms. The results are reported on VOC val set with IoU metric. Our learned representations achieve better unsupervised semantic segmentation.

	No Hierarchy	HSG	KMeans	NCut	FINCH
ms	120	158	165	170	381

Table 3.6: Our method imposes less runtime overhead than other hierarchical clustering methods. All methods are conducted on a 640×640 image, which is hierarchically partitioned into $25, 16, 9$ and 4 segments. While major latency comes from the pixel embedding network, HSG is still 17% faster than KMeans.

iterate KMeans and NCut for 30 times. As shown in Table 3.6, our HSG imposes less runtime overhead than other clustering methods. While major latency comes from the backbone CNN, HSG is still 17% faster than KMeans.

3.5 Summary

We deliver the first unsupervised hierarchical semantic segmentation method based on multiview cosegmentation and clustering transformers. Our unsupervised segmentation outperforms baselines on major object- and scene-centric benchmarks, and our hierarchical segmentation discovers semantics far more accurately.

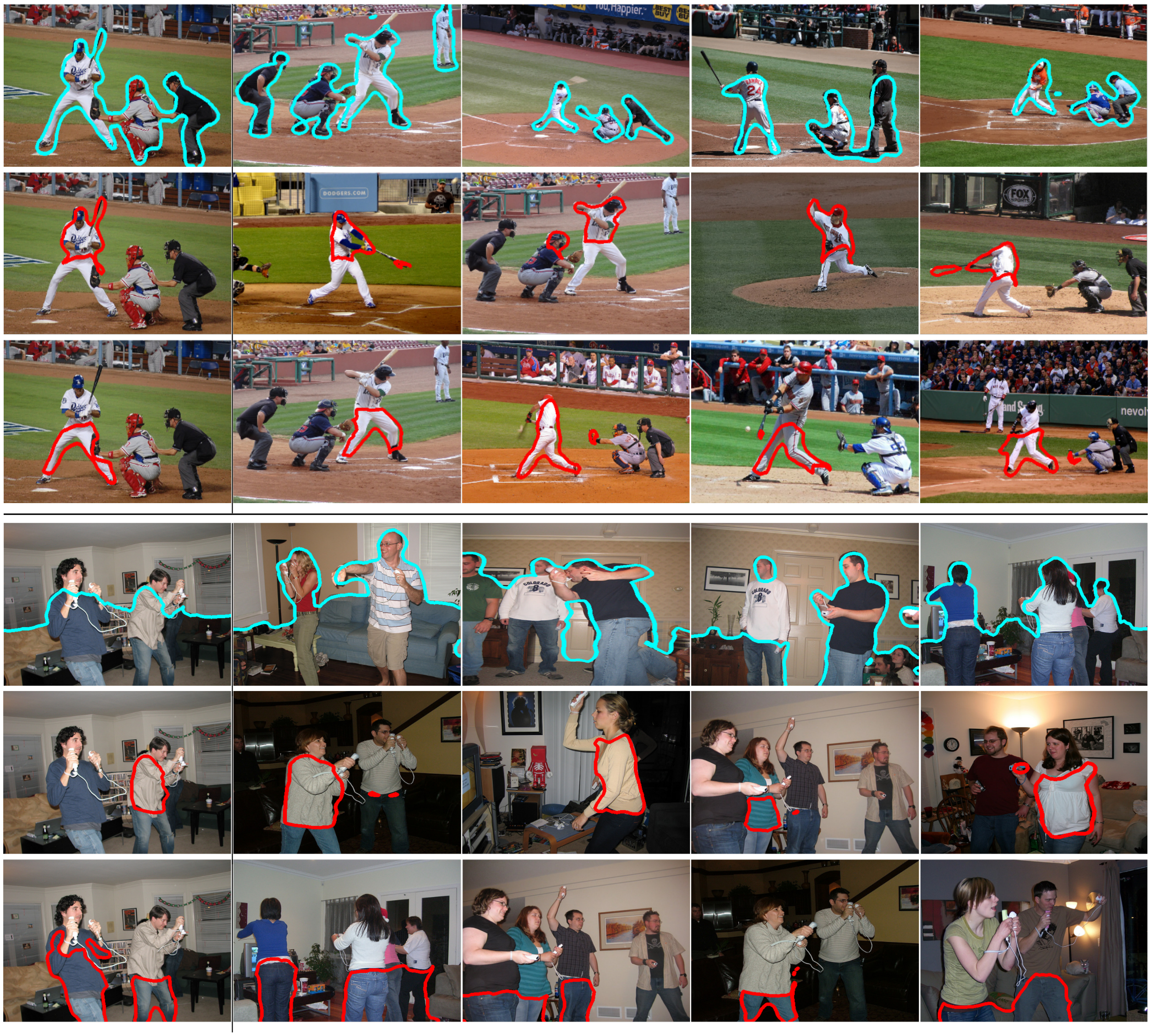


Figure 3.10: Sample retrieval results in MSCOCO for two images, baseball (Rows 1-3) and wii sport (Rows 4-6), based on our CNN features. Column 1 shows a query segment and Columns 2-5 are its nearest neighbour retrievals at the same level of the hierarchy. Segments at a coarser / finer level are shown in cyan (Rows 1,4) / red (Rows 2-3, 5-6). Coarser segment retrievals show that our feature learned from hierarchical groupings are reflective of the visual scene layout (For example, Row 1 all has the 3-person baseball pitching configuration despite drastic appearance variations), whereas finer segment retrievals show that our learned feature is precise at characterizing both the segment itself and the visual context around it (For example, the feature of the query segment (*legs*) in Row 3 is indicative of the pitcher pose on the baseball field). Such a holistic yet discriminative feature representation is discovered in a pure data-driven fashion without any semantic supervision.

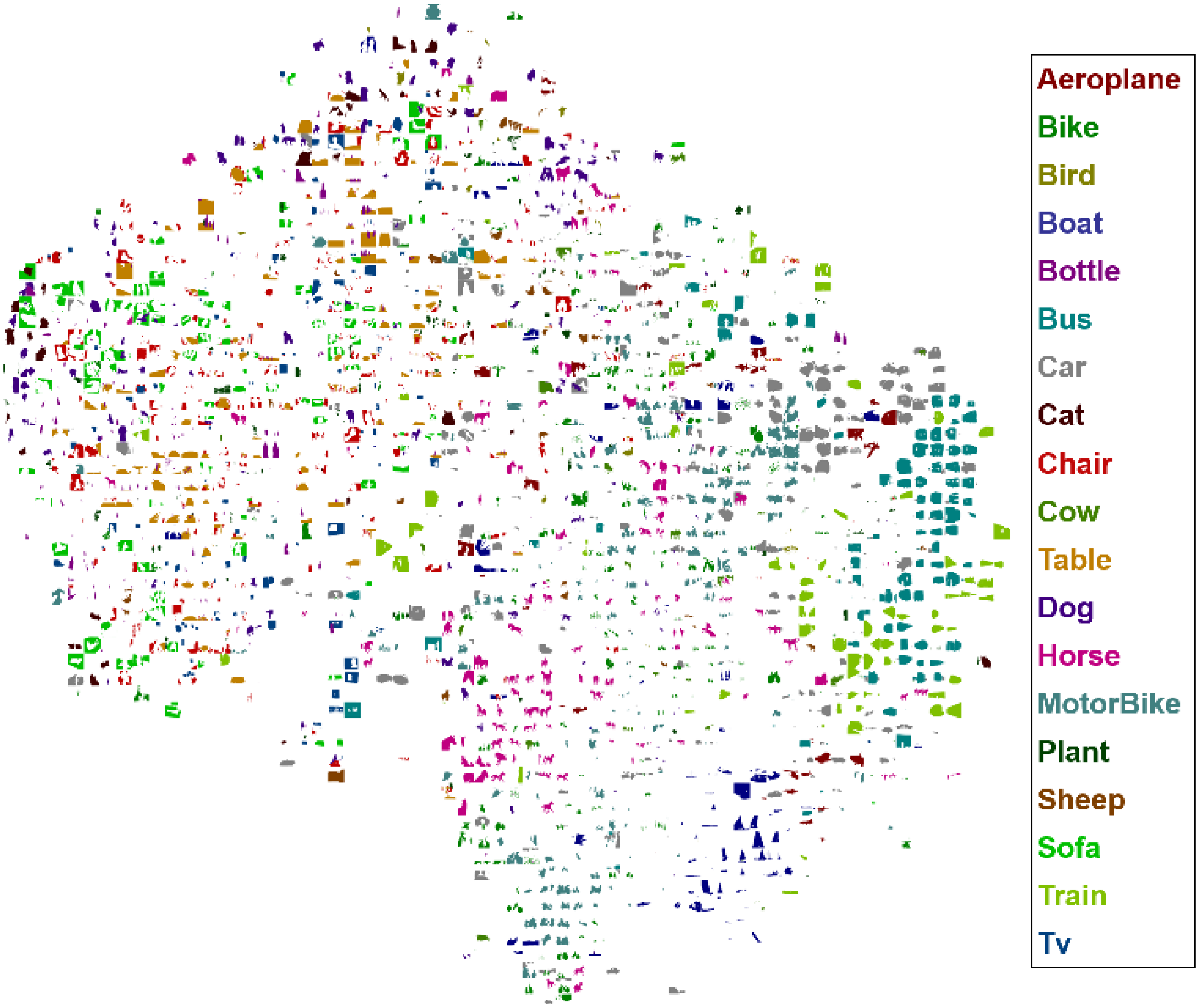


Figure 3.11: Our visual representations encode contextual information of co-occurring objects. We visualize the average feature of *person* category region on Pascal VOC 2012 dataset using tSNE [136]. We use the feature mappings extracted with models trained from scratch on MSCOCO. We represent each *person* category region with the co-occurring object categories, and observe that features in the similar semantic context are clustered.

Chapter 4

CAST: Concurrent Recognition and Segmentation with Adaptive Segment Tokens

4.1 Introduction

Convolutional neural networks (CNN) [99, 2, 81] and Vision Transformers (ViT) [4] have been very successful in computer vision. However, recognizing an image and segmenting it into coherent regions are treated as separate tasks or learned sequentially [18]. Fig. 4.1 illustrates a common practice: CNN (ViT) predicts the semantic class of an image based on the image-level feature from the output of the final convolutional layer (transformer block), and additional clustering based on earlier pixel-wise features is required to generate image segmentation [48, 31].

However, human vision has a general sense of segmentation hierarchy, in terms of groups of pixels or *segments*, before recognition even occurs. This perceptual organization perspective [137, 138] has been overlooked in CNN and ViT architectures: models optimized for image classification tend to latch onto discriminative parts [139] such as faces, often missing inconspicuous body parts that go with the face. Previous methods seldom model how different parts such as *face* and *body* are organized for the whole *animal* explicitly.

To understand the connections between parts and wholes, visual information must be extracted locally and globally. There are three major approaches (Fig. 4.2):

1. **Spatial downsampling:** With pixels laid on a regular grid, features are extracted from patches. The granularity of visual information is determined by the patch size. Max or mean pooling [2, 3], uniform sub-sampling (striding) [81] and patch merging [140] are performed multiple times to increase the effective receptive field size [141]. To generate a segmentation, the output features need to be upsampled and clustered (e.g. K-Means [48]), often resulting in over-smoothed boundaries.
2. **Attention:** Inspired by Natural Language Processing (NLP), image patches are treated

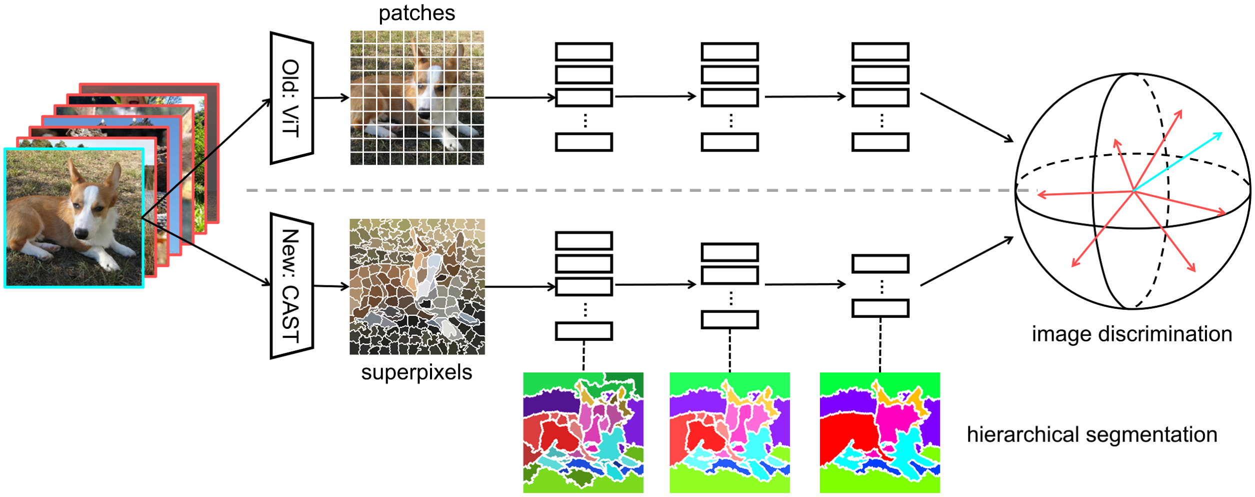
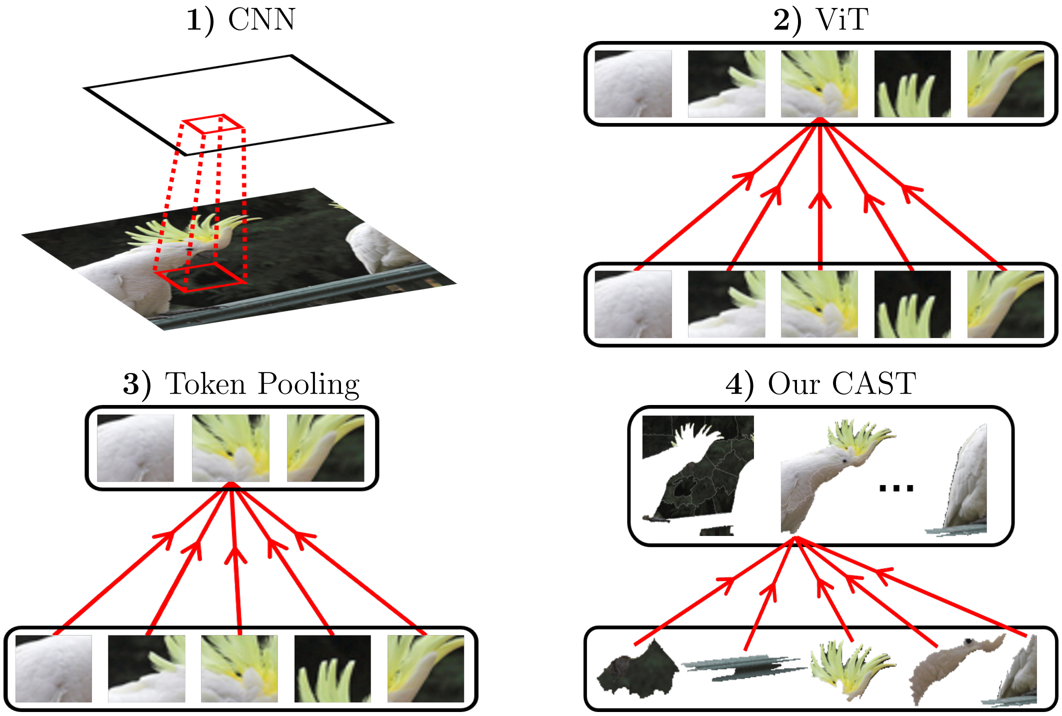


Figure 4.1: We innovate vision transformer models to concurrently learn image recognition and hierarchical image segmentation from unlabeled images alone. **Top:** ViT [4] takes patch tokens as inputs and maintains the same large number of tokens through all encoder blocks. Image segmentation would require additional pixel-wise clustering (e.g. K-Means) on the fixed patch-wise features. **Bottom**: Our model takes segment tokens as inputs and hierarchically groups them into fewer coarsened region tokens. Unlike patch tokens, these segment tokes adapt to the image and vary in shape. We unify fine-to-coarse feature learning at multiple levels in a single model to support not only recognition with maximum image-wise discrimination, but also segmentation with consistency across the hierarchy. Consequently, we achieve better recognition and segmentation with higher computational efficiency.

as visual word *tokens* of the entire image *document*. To extract more global information, ViT contextually updates feature representations based on pair-wise correlation among all the tokens of an image, using attention modules [142]. However, ViT is computationally inefficient as all image tokens are kept in every transformer block.

3. **Significance-based subsampling:** To increase ViT's computational efficiency, tokens are sub-sampled at higher levels based on their significance scores. PoWER-BERT [143] and Token Pooling [144] define the *significance score* as the total attention given to each token from all other tokens. Downsampling then retains only the most dominant visual features in the image. Such methods only keep the most informative tokens in final output representations.

These existing methods have two major issues. **1)** Both CNN and ViT models take regularly shaped patch features as inputs, regardless of what is in the image. Image segmentation derived from such representations often fails to align with contours. **2)** Image segmentation does not involve local-to-global feature extraction, which is treated as a separate visual task from image-wise recognition.



models	CNN	ViT	Token Pooling	Our CAST
nodes	patches	patches	patches	segments
edges	local / fixed	global / dynamic	global / dynamic	global / dynamic
coarsening	✓	✗	✓	✓
segmentation	✗	✗	✗	✓

Figure 4.2: Our model bootstraps from low-level segment tokens, coarsens visual information by clustering, and merges fine-grained segment tokens into coarser-grained region tokens. **Top)** We compare different models in what they operate on and how they extract local-to-global information. CNN [2, 81] computes features on a regular grid and handles more global information by spatial downsampling. ViT [4] takes regularly shaped patches as inputs, and updates features using attention [142]. Token Pooling [144] subsamples tokens by their significance scores. Our CAST takes segment tokens as inputs and coarsens them into larger region tokens, which adapt to the image. **Bottom)** We compare models from a graph perspective, in terms of nodes, edges (connections between nodes), and whether the graph coarsening is used and segmentation is produced. Our method is the only one that uses adaptive segment tokens with coarsening and outputs segmentations.

Our first insight is that pixel groupings are not a computational inconvenience (as opposed to regular patches), but a natural structure to be exploited for better visual computing. Unlike existing CNN and ViT which extract features on a regular grid throughout the entire model, we directly get to low-level pixel groupings at an early stage and develop feature representations subsequently. Our model takes segment features as input tokens and carries this adaptive segment representation through deeper layers. Post-processing with pixel-wise clustering methods is no longer needed.

Our second insight is to derive fine-to-coarse pixel groupings jointly with local-to-global feature extraction. Given a set of token features, we cluster them into fewer components. The next-level feature is the result of pooling current features within each cluster. Since our input tokens come from segments of an image, feature clustering turns fine-grained segments into coarse-grained regions. By repeating the procedure, we obtain a consistent fine-to-coarse (hierarchical) image segmentation and corresponding feature representations at each level of granularity.

We propose to integrate such data-driven perceptual organization into Vision Transformers [4]. We develop *Concurrent Recognition and Segmentation with Adaptive Segment Tokens* (CAST). It has three novel aspects (Fig. 4.2). **1)** We use adaptive segment tokens instead of fixed-shape patch tokens. They no longer live on a regular grid, and their shapes and numbers vary with the image. **2)** We create a token hierarchy by inserting graph pooling between transformer blocks, naturally producing consistent multi-scale segmentations while increasing the segment size and reducing the number of tokens. **3)** We learn segmentation for free *while* training the model for unsupervised recognition by maximizing image-wise discrimination [145]. Neither recognition nor segmentation requires any labeled supervision.

Our experimental results demonstrate our superior computational efficiency and segmentation accuracy on ImageNet and PASCAL VOC. More importantly, our model delivers far more precise foreground masks which can be very useful in a wide range of dense pixel applications.

In short, our work makes three major contributions. **1)** We develop the first vision transformer model that can *concurrently* achieve image-wise recognition and hierarchical image segmentation without any additional processing. **2)** We outperform existing token coarsening methods for both image classification and segmentation tasks. We achieve a better trade-off between model efficiency and task performance. **3)** We deliver better attention maps that capture foreground semantics without supervision, with many potential applications beyond segmentation.

4.2 Related Works

Vision Transformers. Vision transformers (ViT) [4] and its followups [146, 147, 148] adopt the transformer architecture originally for NLP proposed in [142]. ViT achieves remarking performance on image classification [11], however, its high computation costs limit its applications. Its computational complexity is derived from two factors: the latent feature dimensions and the number of tokens.

To reduce the latent feature computation, one direction is to restrict the attention connections by leveraging spatial relationships in the data [149, 150, 151, 152, 153] or utilizing hashing, sorting, or compression [154, 155, 156, 157, 158]. To reduce the number of tokens, two camps of approaches are proposed. The first is to apply the concepts of hierarchical convolutional neural nets to downsample tokens using various pooling methods [140, 159, 160]. The other attempts to measure the significance scores among the tokens and drop or prune tokens accordingly [143, 161, 144]. This camp is the most related to our work. Our work differs in that tokens are not discarded but merged into coarser ones.

TCFormer [162] and GroupViT [163] are two recently proposed hierarchical vision transformers. Both models still use patch tokens. TCFormer is only applied to supervised tasks, and GroupViT requires text supervision. Our work operates on adaptive segment tokens, which naturally induce hierarchical image segmentations. Our model does not require any human label.

Superpixels. Superpixels are referred to as sets of locally connected pixels that contain coherent structures (e.g. colors) [164]. Superpixels have been applied to densely labeling tasks, such as body-part parsing [165], saliency detection [166], image segmentation [167, 168, 169, 170], and hierarchical segmentation [171]. Recently, [172] tackle semantic segmentation by replacing patch with superpixel tokens in ViT architectures. In contrast, our model further creates a segment hierarchy and performs both classification and segmentation concurrently.

Image Segmentation and Clustering. Image segmentation is referred to as partitioning an image into coherent regions. Classic methods have two steps: extracting local features and clustering them based on different criteria, e.g., mode-finding [92, 78], or graph partitioning [93, 94, 95, 96, 97]. A hierarchical segmentation is predicted as output for comparing against human perception [77]. The common approaches, to avoid ambiguities along object boundaries, typically resort to contour detection [98, 76] and eliminate contours iteratively to form multi-scale segmentations [77]. Such approaches train on the finest level of ground-truth segmentation and hope to produce coarser levels of segmentation automatically for inference. Our work operates directly on segments as opposed to contour proxies.

Concurrent Recognition and Segmentation. This idea was explored before the deep learning era: Recognition by grouping compatible patches and segmentation by grouping visually similar pixels are solved together through detected pixel-patch relations, resulting in object-specific segmentation [173, 174] and figure-ground segmentation [175, 176]. These methods rely on not only handcrafted features and grouping cues, but also pre-trained object part detectors, whereas our work does not use any such priors or supervised training.

Self-supervised Segmentation and Representation Learning. Recent works can be categorized into three camps. **1)** A straightforward approach is to leverage self-supervised image recognition and transfer the model to segmentation by increasing the location sensitivity [74, 86, 87, 88], adding a contrastive loss across views [90], or by stronger augmentation and constrained cropping [85, 112]. **2)** A pixel-wise cluster predictor can be learned by maximizing the mutual information between cluster predictions on augmented views of the

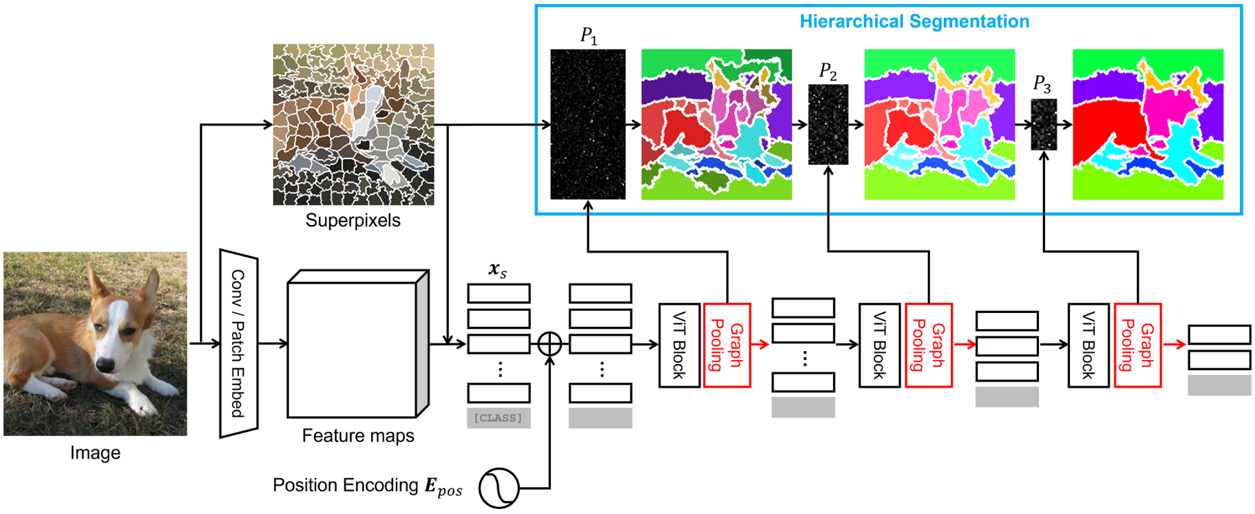


Figure 4.3: Our CAST jointly produces **1) a hierarchical image segmentation** and **2) corresponding multi-level features** for an input image. Building on ViT, our model operates on segment, not patch, tokens. We oversegment the image into superpixels, and extract initial segment tokens $\mathbf{x}_s$. We concatenate $\mathbf{x}_s$ with a `[CLASS]` token, and sum together with position encoding $\mathbf{E}_{pos}$ as inputs to subsequent transformer blocks. Followed by a **graph pooling** module, we group segments into coarser regions and aggregate features within each group. Let P_l indicate how fine segments map to coarse regions at level l. The coarsened tokens become the inputs of next-level encoder blocks. Repeating the procedure, we generate a hierarchical image segmentation and multi-level features.

same instance at corresponding pixels [89, 116]. **3)** A pixel-level feature encoder can be learned directly by maximizing discrimination between pixels based on either contour-induced segments [48], pre-computed region hierarchies [113], or hierarchical clustering transformers [31]. Segmentation is thus derived from pixel feature similarities. Our work unifies the first and the third camp, as we train ViT with a self-supervised image recognition framework while naturally producing unsupervised hierarchical segmentation.

4.3 Concurrent Recognition and Hierarchical Segmentation

At the core of our method is to consider image recognition and segmentation as concurrent, not separate, tasks. The basic idea is to extract local-to-global visual information by producing fine-to-coarse image segmentations and corresponding feature representations. Meanwhile, such multi-scale feature representations should support final image-level recognition.

We ground our approach on a general perspective: images as graphs where pixels are nodes. We first generate low-level superpixels given an input image and extract corresponding segment tokens. Then we integrate image segmentation into the ViT architecture, where transformer blocks and our proposed graph pooling modules take arbitrarily shaped segment, not fixed-shape patch, tokens as inputs and outputs coarsened segment tokens for the next transformer block. In other words, we augment ViT to cluster fine-grained segments into coarse-grained regions w.r.t group-wise correlation in the token feature space, resulting in hierarchical segmentation. By doing so, our model performs classification and hierarchical segmentation concurrently and more efficiently.

We describe the three components in our model in each subsection. **1)** Oversegmentations of input images based on low-level visual cues. **2)** Transformer encoder blocks to update token features. **3)** Graph pooling modules to cluster segment tokens and generate next-level representations. See Fig. 4.3.

The Finest-level Pixel Groupings and Token Features

Existing methods tackle image segmentation separately from feature extraction. Such models extract patch features from the input image and then generate after-the-fact image segmentation by clustering the fixed patch features (similar to SegSort [48] and HSG [31]). In stark contrast, our core idea is to involve image segmentation in feature extraction in the model architecture. We derive pixel groupings at an early stage, where we extract corresponding features for every segment. Our model carries such segment features to the subsequent layers and obtains image segmentations directly from the segment index of each pixel. Post-processing is thus not needed.

Given an input image, we start with the finest-level pixel groupings. We perform pixel groupings based on low-level visual cues to align segmentations with image contours. Specifically, we apply oversegmentation methods, e.g. Seeds [177], to partition an image into locally connected and color-wise coherent regions–superpixels. Detailed in section 4.3, we can produce a hierarchical image segmentation with precisely localized contours by progressively grouping these superpixels.

To extract features of the superpixels, we first convolve the image with multiple convolutional layers, resulting in overlapping patch features. We then average pool those patch features within each superpixel to derive initial segment tokens $\mathbf{x}_s$ (with dimensions of the number of superpixels and number of feature channels).

We then input the initial segment tokens $\mathbf{x}_s$, along with additional priors to our model. Following ViT [4], we append $\mathbf{x}_s$ with a learnable embedding (`[CLASS] token` $\mathbf{x}_{class}$) to encode the most prominent features of an image. $\mathbf{x}_{class}$ is randomly initialized and shared among different input images. We also enforce a spatial prior by adding $\mathbf{x}_s$ with relative position encodings $\mathbf{E}_{pos}$. We initiate $\mathbf{E}_{pos}$ at the same resolution as the convolutional patch features and then average pool within each superpixel. To sum up, the input segment token is $\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_s] + \mathbf{E}_{pos}$.

General and Globalized Backbone Architectures

Most existing vision models digest pixels on a regular grid and update features within a limited range of neighboring grids. Such methods have two limitations: **1)** all features correspond to the same-shape patches in images, and **2)** pixels/patches are locally connected and have little knowledge of global correlation. Yet, segmentations are adaptive to images: every segment has different shapes. Optimal segmentation also requires group-wise correlation among pixels [94]. We thus prefer globalized architectures, such as ViT and GNN [178], over the grid-based models.

Specifically, we select ViT as the backbone, which updates features in the context of all inputs without the need for a regular grid. That is, ViT computes features from a globally-connected graph. It consists of multi-headed self-attention modules (MSA) [142], where features are updated according to pair-wise correlation among all the input features. As a result, ViT encodes global correlation and allows inputs to have different shapes, which is ideal to enable optimal segmentation.

Building upon ViT, our model contains multiple encoder blocks that take segment tokens as inputs: at level l, the encoder block updates its inputs $\mathbf{z}_{l-1}$ to $\hat{\mathbf{z}}_l$. Notably, to extract information at different scales, we vary the shapes and sizes of segment tokens throughout the model. Coarser segmentations have fewer segments (tokens), which improves our model efficiency. For the first block, $\mathbf{z}_0$ corresponds to superpixels. For all the other blocks, $\mathbf{z}_{l-1}$ corresponds to segments at different scales.

Algorithm 3: GraphPool

```
Input: Feature $\hat{\mathbf{z}}$ and number of clusters $n$.
Output: Coarsened feature $\mathbf{z}$ and
  assignments $P$
/* Sample n centroids.                       */
Centroid indices  $S \leftarrow \mathrm{FPS}(\hat{\mathbf{z}})$
/* Refine features to encode
   correlation.                              */
Refined feature  $\mathbf{u} \leftarrow \mathrm{MSA}(\mathrm{LN}(\hat{\mathbf{z}})) + \hat{\mathbf{z}}$
Normalized feature  $\mathbf{u} \leftarrow \mathbf{u} - \mathrm{mean}(\mathbf{u}) + \mathrm{bias}$
Centroid feature  $\mathbf{v} \leftarrow \{\mathbf{u}_i | i \in \mathrm{S}\}$
/* Assign new clustering.                    */
$P \leftarrow \mathrm{softmax}(\kappa \frac{\mathbf{u}\mathbf{v}^\top}{\|\mathbf{u}\|\|\mathbf{v}\|})$
/* Output new features.                      */
Average pooled feature  $\bar{\mathbf{z}} \leftarrow P^\top \hat{\mathbf{z}} / P^\top 1$
New centroid feature  $\mathbf{z} \leftarrow \{\hat{\mathbf{z}}_i | i \in \mathrm{S}\}$
Updated centroid feature
  $\mathbf{z} \leftarrow \mathbf{z} + \mathrm{FC}(\mathrm{LN}(\bar{\mathbf{z}}))$
```

FPS: Farthest Point Sampling.
MSA: Multi-headed Self-Attention.
FC: Fully Connected Layer.
LN: Layer Norm. BN: Batch Norm.

Algorithm 4: Overall framework

```
Input: Initial segment token $\mathbf{x}_s$, class
  token $\mathbf{x}_{class}$, position encoding $\mathbf{E}_{pos}$
  and grouping steps $\Delta$
Output: Feature $\mathbf{f}_{class}$ and $\mathbf{f}_{seg}$
/* Input tokens with priors.  */
$\mathbf{z}_0 \leftarrow [\mathbf{x}_{class}; \mathbf{x}_s] + \mathbf{E}_{pos}$
/* ViT with Graph Pooling.  */
for $l = 1 \ldots L$ do
    $\hat{\mathbf{z}}_l \leftarrow \mathrm{ViT\_Encoder}(\mathbf{z}_{l-1})$
    if $l \in \Delta$ then
        $\mathbf{z}_l \leftarrow \mathrm{GraphPool}(\hat{\mathbf{z}}_l)$;
    else
        $\mathbf{z}_l \leftarrow \hat{\mathbf{z}}_l$;
    end
end
/* Outputs for classification.
   */
[CLASS] token  $\mathbf{f}_{class} \leftarrow \mathrm{LN}(\mathbf{z}_L^0)$
/* Outputs for segmentation.  */
Multi-level segment tokens  $\mathbf{f}_{seg} \leftarrow$
  $\mathrm{FC}(\mathrm{BN}([\mathrm{Unpool}(\mathbf{z}_l^{1:n_l})); \ldots])$   $l \in \Delta$
```

Hierarchical Groupings of Segment Tokens

Starting with superpixels, we group fine-grained segment tokens into coarse-grained region tokens to obtain more global visual information. With fine-to-coarse segment groupings, we can directly induce hierarchical image segmentations of an input image. As we map superpixels into token features, hierarchical segment groupings become a multi-scale feature clustering and pooling problem. We have two considerations: **1)** The number of coarser segments (tokens) should be freely adjustable during training and inference. **2)** The model should achieve optimal partitioning of inputs. We summarize our feature clustering algorithm, dubbed Graph Pooling Module, in Alg. 3.

Existing methods [163, 31] perform clustering with a set of learnable embeddings, resulting in fixed segmentation granularity. Yet, the optimal number of segments varies with image scales: larger (smaller) images require more (fewer) segments. Instead, we conduct feature clustering with an arbitrary number of centroids that are initialized by sampled inputs. The number of centroids corresponds to the segmentation granularity. We apply the Farthest Point Sampling (FPS) algorithm [179] to select a subset of token features as initial centroids. The FPS algorithm enables the sampled centroids to cover the input feature distributions, unbiased w.r.t. dominant features. We can set the number of clusters flexibly during training and inference.

In particular, we predict the soft clustering assignments P_l, which indicates how input features are assigned to sampled centroids. We calculate P_l as softmax-normalized pair-wise similarity among inputs and sampled centroids. We then generate coarsened tokens $\mathbf{z}_l$ by weighted-average aggregating within a cluster. For detailed computation, please refer to Alg. 3.

Training and Inference

We summarize our overall framework in Alg. 4. We conduct segment grouping at certain levels in the model. We train our CAST using an image-wise contrastive learning framework–Moco-v3 [145]. The objective is to contrast each image against others in the latent feature space

To predict classification, we follow MoCo-v3 to apply a 3-layer MultiLayer Perceptron (MLP) head on output `[CLASS]` token. To predict segmentation, we fuse multi-level features as outputs [180] and unpool higher-level features based on the grouping index w.r.t superpixels.

4.4 Experiments

Datasets

ImageNet [11] is an object-centric image classification dataset, annotated with $1,000$ object categories (a.k.a IN-1k). Each image is labeled with one object category, and objects are mostly located at the image center. The training and validation set include 1.28M and 50K images, respectively. Additionally, we follow [181] to subsample 100 object categories to create IN-100. The subset consists of 127K and 5K images for training and testing.
Pascal VOC 2012 [15] is an object-centric semantic segmentation dataset, which contains 20 object categories and a background class. We use the augmented training set [128] with $10,582$ images and validation set with $1,449$ images.
MSCOCO [16] is a generic scene parsing dataset with 80 object categories. The scene contexts are more complex and more objects are included in each image (7.3 vs. 2.3) than VOC. Following [85], we train on $118,287$ images of *train2017* split and test on the VOC validation set.

Architecture and Training

Architecture. For most experiments, we base our architecture on ViT-S [4], which has 384 channel dimensions of all encoder blocks. We follow [182] to **1)** replace the patch-wise linear projection layer with a stack of four 3×3 and one 1×1 convolutional layers; **2)** use 11, not 12, encoder blocks to maintain similar model capacity. We adopt the same design choice for our vanilla ViT baselines. For vanilla ViT (our CAST), we set stride to 2 among the first four (three) convolutional layers. Our models aggregate pixel features within superpixels, resulting in a similar number of input tokens as ViT baselines. We set graph pooling step $\Delta = \{3, 6, 9\}$ and reduce the number of tokens to $\frac{1}{3}, \frac{1}{6}, \frac{1}{12}$ of initial inputs.

K-Medoids clustering baselines. There is no released code for Token Pooling [144]. We thus re-implement the method by combining PoWER-BERT [143] with K-Medoids clustering algorithm. We follow the same settings as our CAST: we adopt segment tokens and use the same number of tokens at intermediate layers as our CAST.

Image resolution and token numbers. For training on all datasets and testing on ImageNet, we set crop_size to 224 and partition 196 superpixels from an image, resulting in around 196 input tokens. For testing on VOC, we generate 1024 (384) superpixels from a 512×512 input image, resulted in 1024 (384) input tokens for semantic (figure-ground) segmentation. For vanilla ViT, we adopt the same image resolution and use 196 and 1024 input tokens on ImageNet and VOC.

Parameter	IN-100	IN-1K	COCO
batch_size	256	256	256
learning_rate	$1.5e^{-4}$	$1.5e^{-4}$	$1.5e^{-4}$
weight_decay	0.1	0.1	0.1
momentum	0.9	0.9	0.9
total_epochs	200	100	400
warmup_epochs	20	10	40
optimizer	Adam	Adam	Adam
learning_rate_policy	Cosine decay	Cosine decay	Cosine decay
MOCO : temperature	0.2	0.2	0.2
MOCO : output_dimension	256	256	256
MOCO : momentum_coefficients	0.99	0.99	0.99
MOCO : MLP hidden dimension	4096	4096	4096

Table 4.1: Hyper-parameters for training our CAST, K-Medoids clustering, and vanillar ViT on IN-100, IN-1K, and COCO dataset. We follow mostly the same set up as MoCo.

Model Training. Based on MoCo, we train all models from scratch without any human-labeled supervision. Mostly follow MoCo (Table 4.1), we set batch_size to 256, learning_rate to $1.5e^{-4}$, weight_decay to 0.1, and momentum to 0.9. We use AdamW [183] optimizer. For hyper-parameters of MoCo framework, we set temperature to 0.2, output dimension to 256, momentum_coefficients to 0.99. The 3-layer MLP head has a hidden dimension of 4096. For IN-100, IN-1k and COCO, we set training epochs to 200, 100 and 400, along with warmup_epochs to 20, 10 and 40, respectively. Cosine decay schedule is applied to adjust the learning rate.

Inference and Testing

On ImageNet, we follow MoCo to apply linear probing to evaluate model performance. We freeze the trained model weights and replace the 3-layer MLP head with a randomly initialized linear projection layer as classifier. We train the linear classifier with ground-truth labels and report Top-1 accuracy. On VOC, we follow [85] to predict semantic segmentation via nearest neighbor search from the labeled VOC training set. We also evaluate transfer learning performance by fine-tuning models on training set and test on validation set. Lastly, we report figure-ground segmentation performance by binarizing multi-head attention maps. Following DINO [184] to generate binary segmentation, we threshold attention maps by keeping only 60% of the mass, and select the best binary segmentation among all attention maps for each image. We next present more details of inference and testing procedure for each task.

Image classification: linear probing. We follow MoCo-v3 [145] to evaluate image-wise discrimination model performance using a linear probing protocol. We freeze the trained model weights and replace the 3-layer MLP head with a randomly initialized linear projection layer as classifier. We train the linear classifier with ground-truth labels and report the top-1 accuracy. Following [145], we train the linear classifier with 90 epochs on ImageNet dataset. We set momentum to 0.9 and weight_decay to 0 for all experiments. On IN-1k, we set batch_size to 1024, learning_rate to 30; on IN-100, we set batch_size to 256, learning_rate to 0.8. SGD is used as the optimizer.

Semantic segmentation: segment retrieval. We follow [48, 85, 31] to evaluate semantic segmentation using segment retrieval. We partition an image into several segments, and conduct nearest neighbor search to predict the label for each segment. We assign the majority labels from 20 retrieved segment.

For ViT baseline, we apply the MLP head on each token to generate unit-length output features, and upsample the feature maps to the original resolution of the input image. Followed by spherical K-Means clustering algorithm, we partition the image into 36 segments using the output features.

Our CAST does not require additional upsampling and K-Means clustering. For segmentation, our model follows Hypercolumn design [180] to unpool and fuse multi-level segment tokens. Our model generates the same number of outputs tokens as the superpixels. We gather pixel features from output tokens based on the superpixel index. Without the need of spherical K-Means clustering, our CAST predicts an image segmentation using the graph pooling modules. We compute normalized segment features according to such image segmentation.

Semantic segmentation: transfer learning. We follow [85] to evaluate model performance using transfer learning protocol. All models are unsupervisedly trained on MSCOCO, and supervisedly fine-tuned on Pascal VOC. We replace the 3-layer MLP head with 2-layer 1×1 convolutional layers. We set the training steps to 30K, learning_rate to 0.003, weight_decay to 0.0001, batch_size to 16, crop_size to 512. Following [80], we adopt poly learning rate policy by multiplying base learning rate by $1 - \frac{iter}{max\_iter}^{0.9}$. We adopt SGD optimizer. Only single-scale image is used for inference.

backbone	method	# tokens	GFLOPS	IN-100	IN-1k
ViT-S	Vanilla	$[196] \times 4$	4.67	78.1	**67.9**
	K-Medoids	$196, 64, 32, 16$	**2.84**	75.8	62.8
	Our CAST	$196, 64, 32, 16$	3.12	**78.9**	66.1
ViT-B	Vanilla	$[196] \times 4$	17.8	81.7	-
	K-Medoids	$196, 64, 32, 16$	**10.8**	81.6	-
	Our CAST	$196, 64, 32, 16$	11.7	**82.0**	-

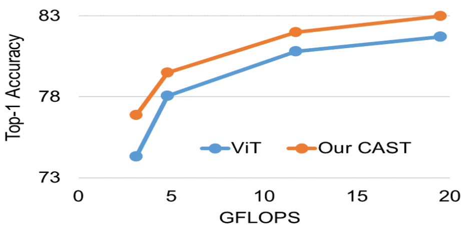


Table 4.2: Our model achieves a better trade-off between model efficiency and task performance for unsupervised image classification on ImageNet val set. We report the top-1 accuracy of the linear classifier. **Left:** Image classification on IN-100 and IN-1k. Compared to vanilla ViT, our CAST reduces computation overhead by decreasing the number of tokens in the intermediate blocks. **Right:** Image classification on IN-100 val set with different model sizes. Our CAST outperforms vanilla ViT using the same computational budget.

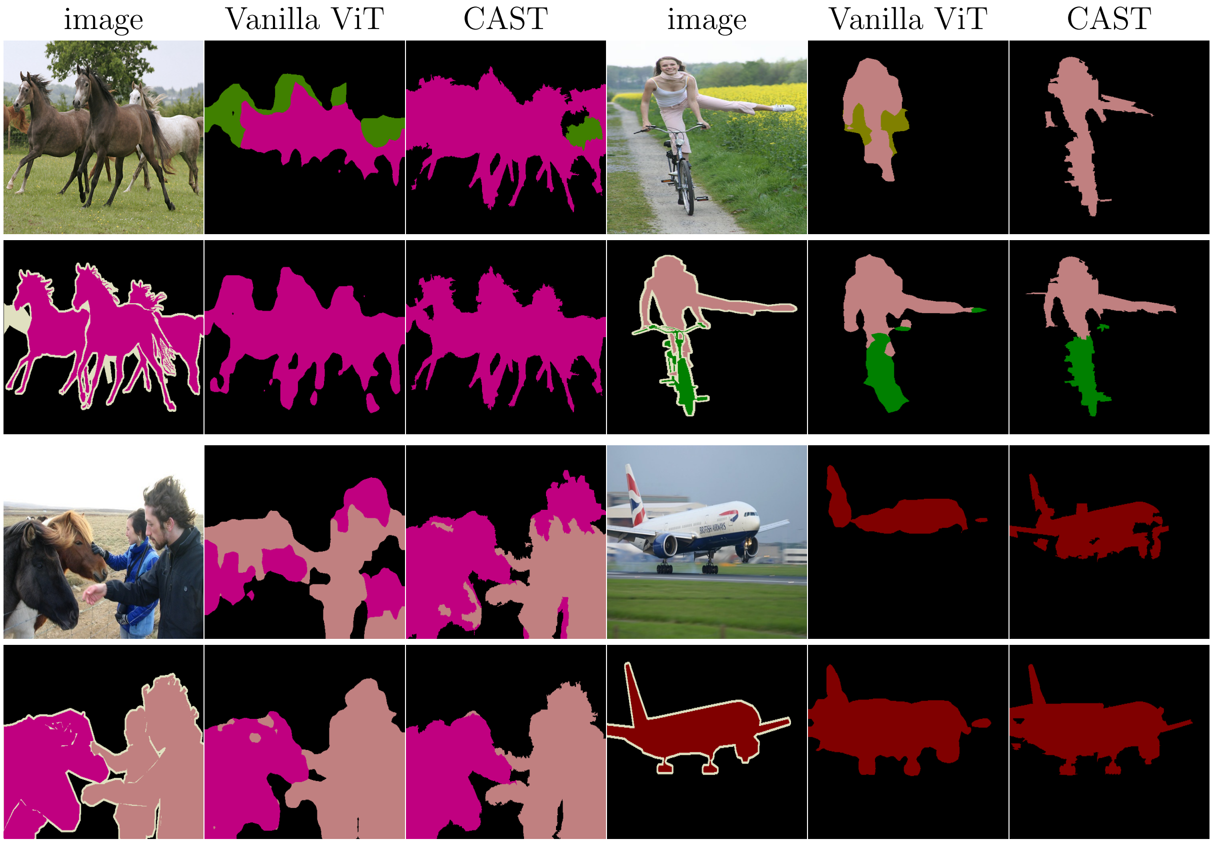
Quantitative Results on Unsupervised Image Classification

Our model achieves a better trade-off between model efficiency and task performance for unsupervised image classification. We report top-1 accuracy on both IN-1k and IN-100 (see the left of Table 4.2). On IN-100, using ViT-S and ViT-B backbone, our CAST achieves comparable performance as vanilla ViT, yet our model is 66.8% and 65.7% computationally more efficient. On IN-1k, our CAST maintains 97.3% performance of ViT baseline (66.1% vs. 67.9% accuracy). Our graph pooling module consistently outperforms K-Medoids clustering. Due to the computation limitations, we set smaller batch_size and training epochs on IN-1K. We also do not apply the pre-training [144] or model distillation [161] strategy. We can further improve the performance based on such settings.

On IN-100, our model outperforms baselines under different model sizes (see the right plot of Table 4.2). Our CAST outperforms vanilla ViT using the same computational budget.

Results on Unsupervised Semantic Segmentation

We evaluate unsupervised semantic segmentation on VOC in Table 4.3. We report the regional mean Intersection of Union (mIoU) and boundary F-measure metric. For both segment retrieval and transfer learning, our segmentations are more precise (+5.0% and +1.0% mIoU) and respect object boundaries better (+8.6% and +7.4% F-measure) than vanilla ViT, yet model efficiency is greatly improved. Notably, without applying CRF for post-processing, our model is still able to preserve thin structures (e.g. horse legs) well, whereas ViT results are over-smoothed.

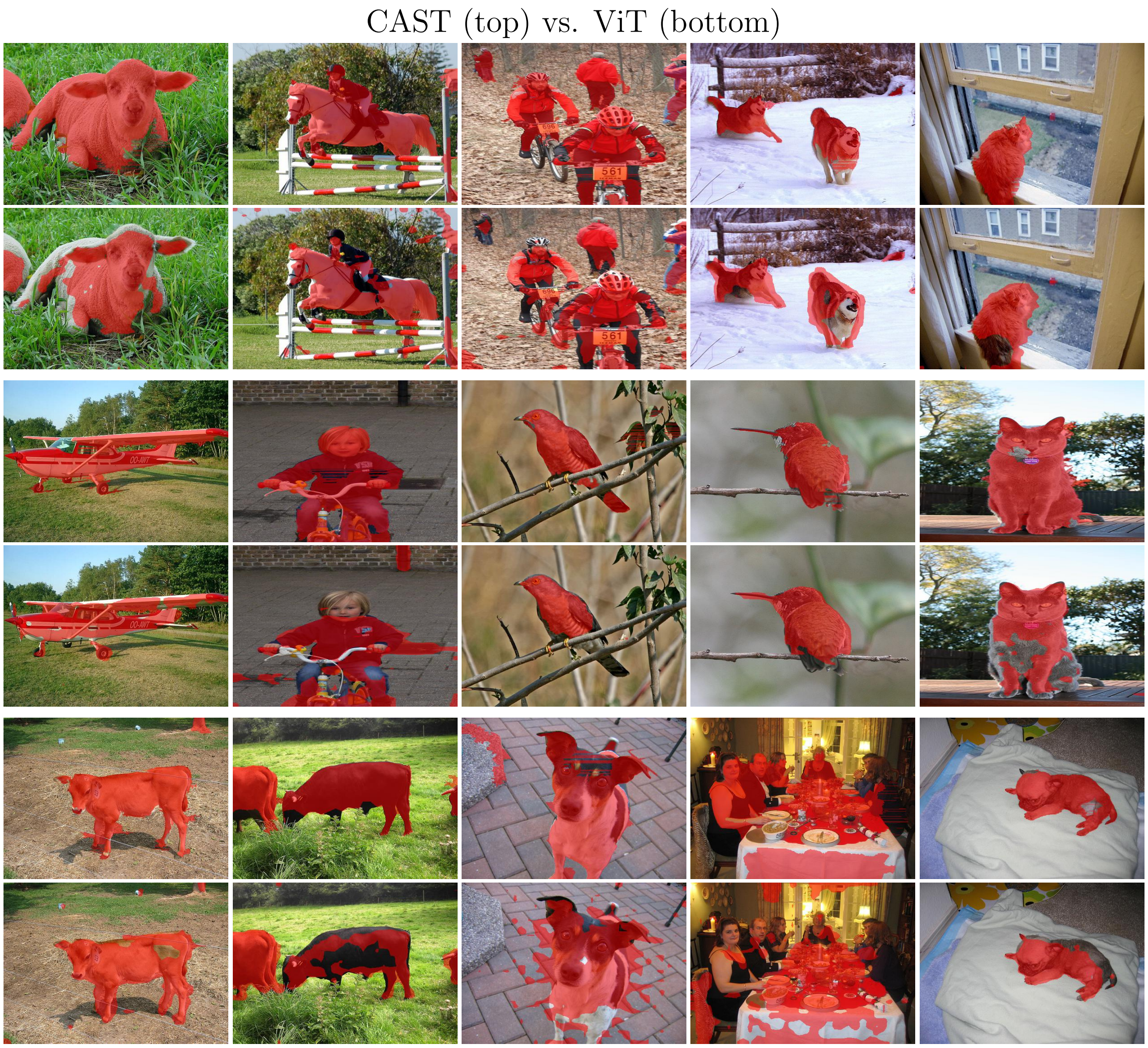


Trained on MSCOCO, to be tuned on VOC			before		after	
method	# tokens	GFLOPS	mIoU	F-score	mIoU	F-score
Vanilla ViT	$[1024] \times 4$	34.1	30.9	16.1	65.8	40.7
K-Medoids	$1024, 320, 160, 80$	**21.2**	27.6	17.2	66.7	47.5
Our CAST	$1024, 320, 160, 80$	23.6	**35.9**	**24.7**	**66.8**	**48.1**

Table 4.3: Our predicted segmentations are much more precise and better aligned with ground-truth (row 2 and 4 column 1 and 3 image) semantic boundaries on VOC val set, benchmarked with the regional mIoU and boundary F-score metric. Segmentations are predicted based on segment-wise nearest neighbor retrievals (before fine-tuning, white-colored table columns, row 1 and row 3 images) and fine-tuned models (after tuning with supervision, gray-colored table columns, row 2 and row 4 images). Our model achieves better image segmentation with less computation.

Results on Attention-induced Figure-ground Segmentation.

We show that our latent attentions capture semantics more precisely. We report the Jaccard similarity between ground truths and predicted foreground masks. Summarized in Table 4.4,



CAST (top) vs. ViT (bottom)

ViT-S/16: foreground prediction	#epochs	batch size	IoU (%)
Our CAST	100	256	48.0
Vanilla ViT	100	256	45.8
DINO	300	1024	45.9

Table 4.4: Our CAST attends to foreground semantics more precisely. All models are trained on IN-1K from scratch. Due to computation limitations, we train vanilla ViT and our CAST with smaller batch size and fewer epochs. **Top)** Visual comparison between our CAST (top row) and vanilla ViT (bottom row). Our predicted masks are more coherent within foreground regions and better align with object boundaries. **Bottom)** Jaccard similarity between ground-truth and predicted foreground masks.

our CAST outperforms DINO and ViT by +2.1%. Our foreground masks can cover dominant foreground objects and better align with object boundaries.

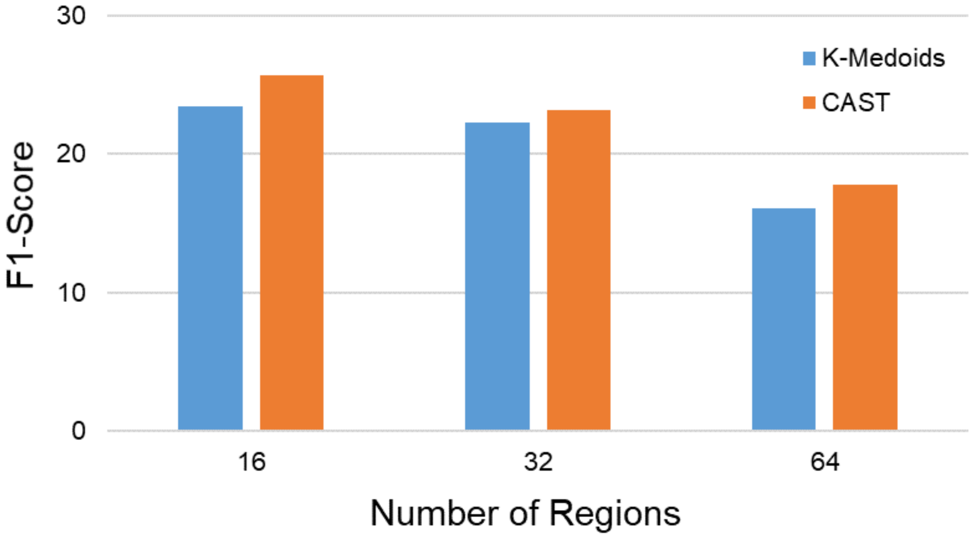


Figure 4.4: Our CAST delivers better hierarchical segmentation than K-Medoids clustering algorithm. On DensePose [29], we parse body-part-level labels into head, torso, upper and lower limb regions. Using CAST or K-Medoids, we partition an image into 64, 32 and 16 regions, and evaluate the region-wise converings with ground-truths by F1-score. Our CAST captures semantics better at every level of granularity.

Quantitative Results on Hierarchical Segmentation

To demonstrate the efficacy of our hierarchical segmentation results, we compare CAST with K-Medoids on the DensePose dataset [29]. We process body-part-level labels into the head, torso, upper limb, and lower limb regions. We segment an image into 64, 32, and 16 regions, and evaluate the region-wise coverings [77] with ground truths by F1-score. As shown in Fig. 4.4, our CAST outperforms K-Medoids and is better at capturing semantics at every level of granularity.

Visual Results on Hierarchical Segmentation

We present more visualization results of hierarchical segmentations induced by vanilla ViT, K-Medoids and our CAST (see Fig. 4.5). We segment the input image into 32, 16 and 8 regions, hierarchically. Notably, ViT requires additional pixel-wise K-Means clustering on fixed feature representations, yet the results are over smoothed. Our CAST naturally

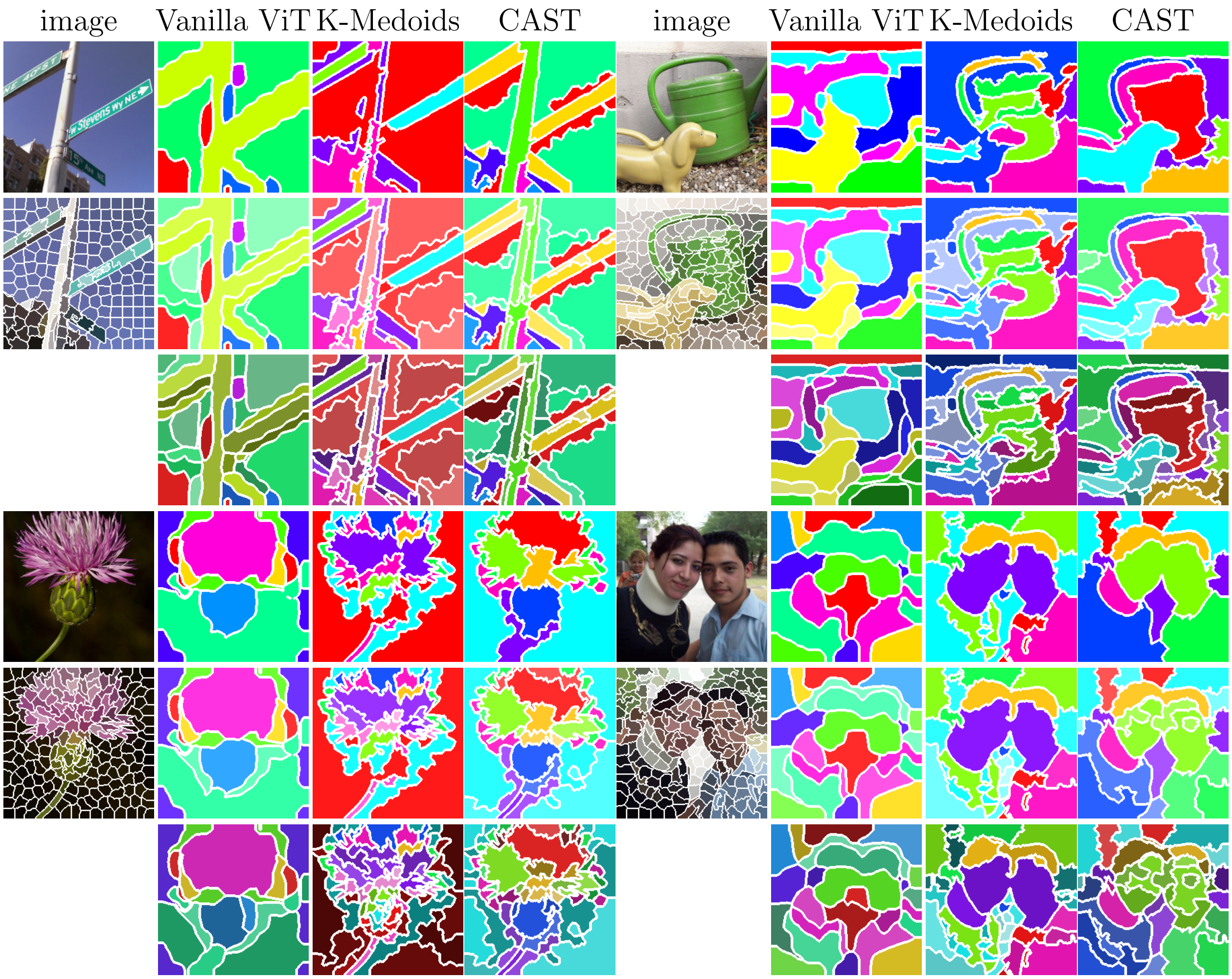


Figure 4.5: Our CAST generates higher-quality hierarchical segmentation. From left to right, we show the input image and hierarchical segmentations generated by vanilla ViT, K-Medoids clustering, and our CAST. We also show the superpixels (white contours) generated from input images. From top to bottom, we hierarchically segment images into 8, 16, and 32 regions. ViT requires additional pixel-wise clustering (e.g. K-Means) on fixed features, yet the results are over-smoothed. Our CAST naturally generates hierarchical segmentations which capture semantics more precisely.

generates hierarchical segmentations without any post-processing. Our segmentations better align with image boundaries and capture semantics more precisely at each level of granularity.

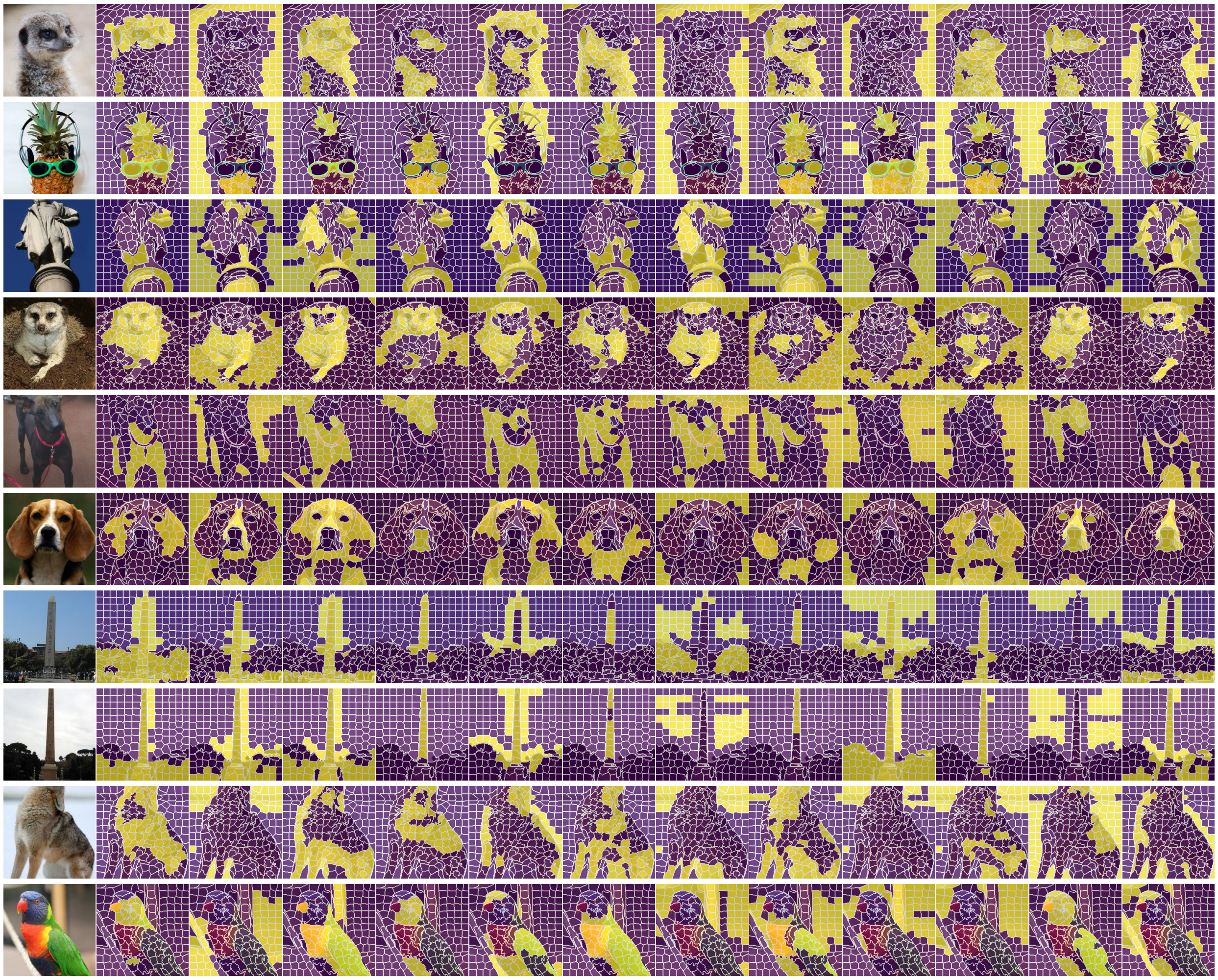


Figure 4.6: Our multi-head attention maps reveal parts-of-the-whole information of the image on IN-100. **From left to right:** input images and corresponding 12 heads of attention maps of the [CLASS] token to all the other segments. We follow DINO [184] to binarize attention maps. We show that same object parts are together attended in the same head, e.g. face vs. ears vs. nose of the dog. Our model takes segment tokens, resulting in attentions better aligned with object boundaries.

Visual Results on Multi-head Attention Maps

We visualize the multi-head attention maps of the [CLASS] token to all the other segment tokens in our vision transformer. As the [CLASS] token is optimized for image-wise discrimination, such attention maps indicate the most informative groupings of segments that will induce the most discriminative image-wise representations. We visualize the same attention maps used to generate the figure-ground segmentation, which are the ones in the 9^{th}

transformer encoder block. The layer takes 32 coarsened segment tokens as inputs, resulting in 12 heads of 32 × 32 attention maps. We follow the same procedure as DINO [184] to display the binarized attention maps. The threshold is adjusted to keep 60% of the mass. See [184] for more details.

As shown in Fig. 4.6, our attention maps reveal parts-of-the-whole information of the image. We observe that the same object parts are together attended in the same attention head, e.g. face vs. ears vs. nose of the dog. It indicates that image-wise recognition requires parts-of-the-whole information. Additionally, our model carries segment, not patch, tokens through the layers, resulting in attention maps better aligned with object boundaries.

Token	Pooling	Acc.
Patch	-	78.1
Segment	-	78.1
Segment	✓	78.9

Pooling	Acc.
Our Graph Pooling	78.9
K-Medoids: PoWER-BERT	75.8
K-Means: PoWER-BERT	73.9
K-Medoids: Random Sampling	72.3

Table 4.5: Our proposed graph pooling module improves image classification on IN-100 val set. **Left:** Improved performance by adding our graph pooling module. **Right:** Our graph pooling module outperforms K-Medoids and K-Means clustering algorithm by large margin. Cluster centroids are initiated by either PoWER-BERT or random sampling.

Ablation Study of Proposed Graph Pooling Module.

Summarized in Table 4.5, we demonstrate the efficacy of the proposed graph pooling module. We report top-1 accuracy results on IN-100 dataset. We show that we improve our model's performance by adding the graph pooling module. In addition, our graph pooling module outperforms K-Medoids and K-Means clustering by a large margin.

#. of Tokens	Encoder Blocks	GraphPool (FPS)
196	86.43	63.02 (37.64)
64	25.4	18.2 (9.7)
32	12.9	9.6 (3.0)
16	5	6.1 (1.5)

Table 4.6: FPS in our graph pool module requires additional computation. We report inference time (ms) of each module with 384 channel dimension and 256 batch size on IN-100. Optimizing the token sampling technique is our future work to further increase our model efficiency.

Ablation Study of Inference Latency

We present the comparison of inference latency among our CAST and vanilla ViT architectures. We report the inference time (ms) of each module with 384 channel dimensions and 256 batch sizes on ImageNet-100. As summarized in Table 4.6, our graph pool module with FPS indeed requires additional computation. However, our method can also reduce inference time by decreasing the number of tokens in deeper layers.

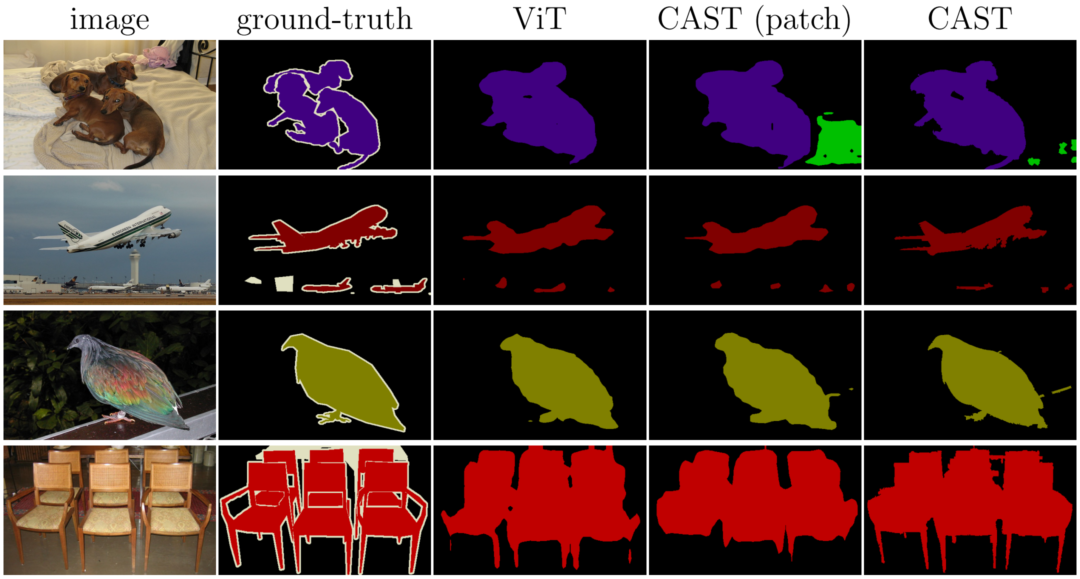
Disregarding the Conv Stem, vanilla ViT and our CAST take 316.91 and 220.57 ms for inference, respectively. Our model is 30.4% faster than ViT. Optimizing the token sampling technique to increase model efficiency is our future work.

Ablation Study of Superpixel

We next verify the superior efficacy of superpixel tokens on dense pixel applications. As shown in Table 4.7, we compare patch tokens to superpixel tokens on image classification and semantic segmentation tasks. For both ViT baseline and our CAST, we observe significant performance gain for semantic segmentation, yet the performance gap for classification is negligible. We conclude that using superpixels can be very useful in a wide range of dense pixel labeling tasks.

4.5 Summary

We develop a novel vision transformer that performs image-wise recognition atop of consistent hierarchical image segmentation, by learning fine-to-coarse features over adaptive segment tokens instead of regular patch tokens. We deliver the first concurrent recognition and hierarchical segmentation model without any supervision, achieving better accuracy and efficiency. The idea can be extended to supervised image classification, with hierarchical semantic segmentation for free.



				Classification	Segmentation	
Method	GFLOPS	dim	Token	Acc.	mIoU	f-score
ViT	65.4	384	Patch	78.1	65.8	40.7
			Segment	78.1	66.5	46.7
Our CAST	42.7	384	Patch	78.1	66.3	41.4
			Segment	78.9	66.8	48.1

Table 4.7: Using superpixel tokens improves performance of dense pixel applications significantly. **Top:** Visual examples of semantic segmentation by fine-tuned models on VOC val set. **Bottom:** Quantitative results on classification and semantic segmentation tasks. We compare patch tokens to superpixel tokens on image classfication (IN-100) and semantic segmentation (VOC). We report the mIoU and boundary f-score performance of semantic segmentation under the setting of transfer learning. We report the GFLOPS of segmentation models, where classification models use the same number of channel dimension. For both ViT baseline and our CAST, we observe significant performance gain for semantic segmentation, though the performance gap for classification is negligible. Superpixel-based methods preserve thin structures of objects much better than their patch-based counterparts.

Chapter 5

Contextual Visual Feature Learning for Zero-Shot Recognition of Human-Object Interactions

5.1 Introduction

Real-world visual perception of an object is far more complex than its own semantic categorization. What surrounds the object has a great impact. For example, drivers pay more attention to pedestrians hustling through an intersection than trolling down a sidewalk; A baby holding a *knife* versus a *bottle* would be seen *and* reacted differently by their caregivers. That is, real-world object recognition is not about attaching class labels to individual objects in isolation, as studied in computer vision recognition benchmarks nowadays, but about recognizing objects *along with* their contexts.

Visual context has been conventionally characterized by statistical co-occurrences of patches and objects, although its definition varies with different formulations: It has been modeled as spatially organized image feature (e.g., *scene gits* [185]), co-occurring object semantics [21, 22, 186, 187, 188], instance statistics [23], or co-occurring instance graphs [24].

Higher-level visual tasks naturally require the differentiation of visual contexts. In human-object interaction (HOI) detection [20, 189, 190], action recognition [191, 192], or scene graph generation[28], semantic classes are defined not just based on the collection of objects themselves, but their poses and relationships with each other: A *person* could *push* a *bike*, *ride* a bike, or *lean against* a bike. In image captioning [193, 16] and visual question answering [194, 195], co-occurring statistics is further refined to reflect that interesting events (not *any* events) are more likely to be named. To understand a movie [196], object relationships are extensively reasoned spatio-temporally and semantically.

One way to learn contextual relationships is from annotations. While annotating the semantic category of objects is time-consuming but not infeasible, annotating visual contexts quickly becomes impractical with an increasing number of object categories. For example, Visual Genome [28] has $33,877$ objects and $42,374$ *pair-wise* object relationships alone. With group-wise or spatio-temporal contexts, the actual number of relationships explodes exponentially. It is not only hard for humans to annotate, but also ineffective for models to predict a large set of contextual relationships.

Our key insight is to approach visual context as a representation learning problem, *not* a classification problem [48, 30]. Instead of predicting discrete relationship categories, we learn to map object instances of similar (*dissimilar*) contexts close (*far*) in some feature space. Where an object instance is located in the feature space indicates the type of visual context it belongs to. Such a representation learning model is infinitely scalable, unconstrained by the total number of objects or relationships.

We demonstrate the above concept by learning contextual visual features for recognizing human-object interactions without using any annotations on such relationships. Existing annotated interactions only consider a restrictive subset of object-pair relationships, e.g., the pairwise relationship of *a person riding a horse* detected on green grass vs. a soccer field carries different perceptual qualities. We therefore use large-scale generic image datasets such as MSCOCO [16] as the training set, although they are only annotated with object instances and their semantic categories [17].

We model visual context in terms of spatial configuration of semantics between objects and their surrounds, and train their feature representations in a contrastive fashion accordingly. We use a convolutional neural network (CNN) to learn a pixel-wise feature mapper that encodes visual information centered at each pixel semantically and spatially. Pixels that are closer in the feature space have not only similar visual appearances in the same semantic category, but also similar spatial arrangements of surrounding semantics. For example, *a person riding a horse on grass*, *a person riding a horse on street*, *a person walking a horse on grass* should form their individual clusters instead of being mixed up in one cluster.

We formulate a pixel-to-segment contrastive learning loss [48] for *contextual visual feature learning*, where pixels are attracted to their positive segments and repelled from their negative segments. The positive and negative segment sets for each pixel are defined based on not only its own instance and semantic information [111], but also its surrounding semantics.

Visual contexts emergent in such contrastively learned features are completely data-driven and more general than supervisedly learned models. Benchmarked on HICO [189] for recognizing human-object interactions for each person instance, our unsupervised model trained only on MSCOCO with annotations of semantics not relationships outperforms the basic supervised relationship classifier and approaches the state-of-the-art supervised model, both specifically trained on HICO relationships! In addition, we show that unsupervised characterization of visual context helps learn more discriminate features that can improve semantic segmentation performance.

5.2 Related Work

Instance context. Earlier works studied instance contextual relationships mainly to improve object detection. [24] proposed an instance-wise exemplar and 2D spatial graph to model context. [22, 188] and [186, 187] proposed Hand-crafted features and tree-based models to capture context in terms of co-occurring statistics and spatial configurations among objects and their semantics [23]. Recent works have developed graphs [197] or spatial memory [198] to encode context in their deep learning models. We instead capture context implicitly in our learned feature space, and remarkably, we are able to recognize high-level contextual relationships (e.g., human-object interactions) automatically..

Human-object interaction. Since various Human-Object Interaction detection works constructed large-scale labeled image datasets [20, 189, 19, 199], significant progress has been achieved for this problem with different methods such as box transformations [20, 25, 200, 201], two channel interaction [189, 202], mutual contexts of human pose and object [200, 203, 204], Contextual correlation [205], correlation prior of interactions [206], Visual transformer [207, 208, 209, 210], or Graph modeling [211, 26, 212, 213, 214]. Beyond typical human and object appearance features, in order to improve the generalizablilty of the relationship detection, HOI detection works devise various information as input such as human pose [200, 215, 216] or linguistic prior knowledge [217, 211, 218, 219, 220], which require extra human labeling effort to capture such knowledge. More recent works combine these various cues [200, 216, 221, 222]. Our model does not require additional cues such as human pose or language other than RGB images.

Weakly-supervised [223] and zero-shot relationship detection [224, 225, 226, 219, 221, 227] have been studied to improve data efficiency. However, existing zero-shot learning works require large-scale external data to pre-train linguistic knowledge. Although [206, 228] uses prior knowledge that can be obtained from the target training data itself, their method shows limited generalizability on unseen image domains. Weakly-supervised learning still requires laborious image-level annotations.

In contrast, our unsupervised learning is more general: It requires neither target domain information nor relationship labels when training on source data such as MSCOCO. That is, our unsupervised visual context predictor delivers better zero-shot performance than the supervised counterpart!

5.3 Contextual Visual Feature Learning without Supervision

We approach contextual relationship recognition as a feature learning problem. We map pixels to points in a feature space, such that object instances are grouped (separated) if they have similar (different) contextual relationships. Our model does not use or output any pre-defined relationship categories; it groups objects according to their own semantics *and* visual contexts. If a relationship label is desired, we retrieve nearest neighbours of a query in the feature space and transfer their labels.

Unlike supervised learning methods that train a model based on annotated (restrictive) relationships, e.g., *a person riding a bike*, our unsupervised relationship learning method trains a model based on the semantic category distribution at surrounding neighboring patches of the centered object instance.

Our Task: Unsupervised Visual Relationship Learning

Supervised setting. Given an image and a set of detected objects, visual relationship labeling [20, 189, 28] infers the relationship among object instances. Supervised methods [25, 207, 212] can only reason in restricted terms specified by training labels, e.g., between *a pair of objects*. To understand the relationship among *a group of objects*, higher-order information needs to be further extracted.

Unsupervised setting. In a stark contrast to these existing methods, we consider a more general but unsupervised learning setting. We assume no prior knowledge of relationship categories. We train our model on a generic image dataset, given only semantic and instance labels on pixels. Our goal is to infer the relationship of each object instance in a test image. For simplification, we detect object instances using off-the-shelf detectors or ground-truth bounding boxes. For inference, we extract features within an object's bounding box, retrieve their nearest neighbors from a labeled set, and predict relationships by transferring neighbors' labels (see Fig. 5.1).

Evaluation metric. We evaluate the retrieval performance based on the interpolated average precision (AP) metric [229, 15]. We calculate recall (R) and precision (P) by comparing the query's label to the retrieved ones. AP measures the interpolated area under the PR-curve, and is commonly adopted for instance detection and segmentation tasks [16]. See [15] for more details.

Our framework: Pixel-to-segment Contrastive Learning

SegSort [48] is an end-to-end feature learning framework that learns pixel-wise features and the corresponding segmentation based on EM-optimization that maximizes the discrimination among image segments from the entire dataset.

Specifically, a CNN ϕ maps image I to pixel-wise features $V = \{\boldsymbol{v}_i\}$, where $\boldsymbol{v}_i = \phi(x_i)$ denotes the unit-length features centered at pixel x_i. When V is fixed, SegSort generates an image segmentation using the spherical K-Means algorithm [230]. The E-step assigns pixels to their nearest segments. The M-step updates segment features $U = \{\boldsymbol{u}_s\}$ as the length-normalized average pixel feature within each segment: $\boldsymbol{u}_s = \frac{\sum_{i \in R_s} \boldsymbol{v}_i}{\|\sum_{i \in R_s} \boldsymbol{v}_i\|}$, where R_s is the area of segment s.

Let $S = \{s\}$ be the set of segments and z_i the segment index of pixel i. The posterior probability of pixel i belonging to segment s is formulated as: $p(z_i = s | \boldsymbol{v}_i, U) = \frac{\exp(\kappa \boldsymbol{u}_s^\top \boldsymbol{v}_i)}{\sum_{t \in S} \exp(\kappa \boldsymbol{u}_t^\top \boldsymbol{v}_i)}$,

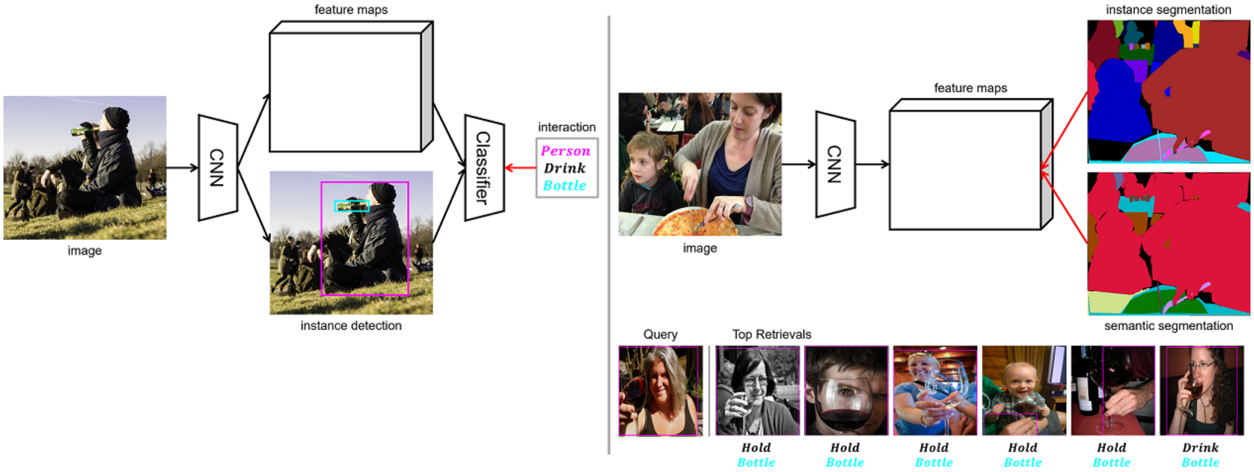


Figure 5.1: Our framework can discover high-level visual contextual relationships automatically. Take recognition of human-object interactions for example [189], interaction labels are composed of (**person**, **interaction**, **object**) triplets. **Left:** Supervised frameworks consider the task as a discrete classification problem. They can only reason in restricted terms specified by training labels, e.g., between *a pair of objects*. **Right:** Our framework tackles the task as a feature learning problem. We learn the feature mappings from semantic and instance labels on pixels. Without any prior knowledge of relationship categories, we predict the interactions of the query person subject by transferring nearest neighbors' labels. Red arrows indicate loss signals.

where κ is the concentration hyper-parameter. To increase the discrimination among segments, pixel features are optimized to minimize the corresponding negative log-likelihood loss: $-\log p(z_i = s|\boldsymbol{v}_i, U)$.

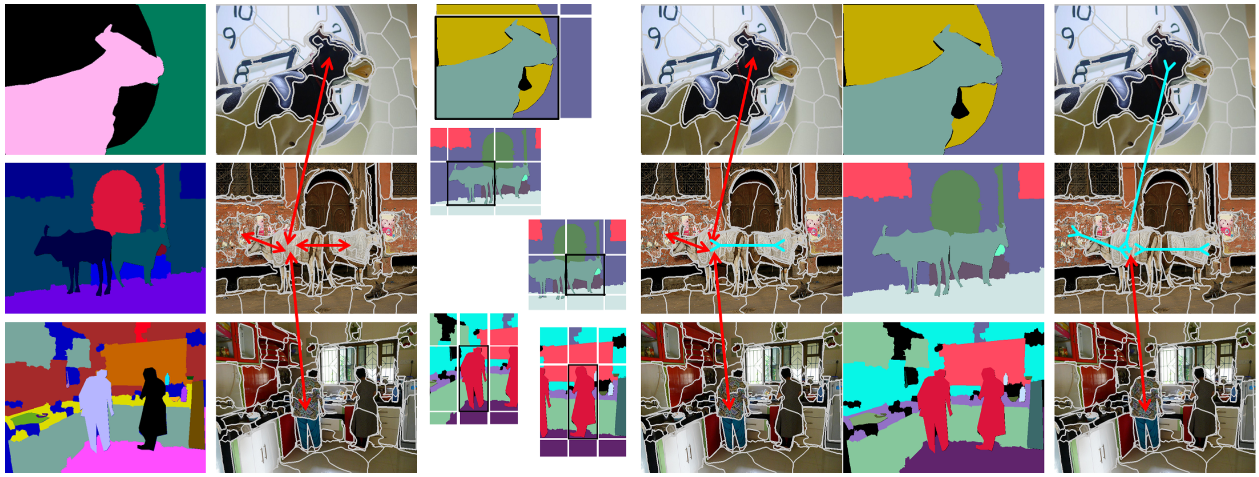
When ground-truth labels C are provided, SegSort adapts the loss in a soft neighborhood assignment formulation [79] to enhance groupings of same-label segments. The pixel-segment contrastive loss is:

$$L(C) = -\log \sum_{s \in C_i^+} p(z_i = s|\boldsymbol{v}_i, U) = \frac{\sum_{s \in C_i^+} \exp(\kappa \boldsymbol{u}_s^\top \boldsymbol{v}_i)}{\sum_{t \in C_i^+ \cup C_i^-} \exp(\kappa \boldsymbol{u}_t^\top \boldsymbol{v}_i)} \tag{5.1}$$

where C defines the positive (negative) set C_i^+ (C_i^-) for pixel i. C_i^+ includes all same-label segments except i's own segment, and C_i^- denotes the set of different-label segments.

Our Loss: Contextual Visual Feature Learning

The ideal contextual feature mapper should capture not only the visual appearance of the object itself, but also the statistical distribution and spatial organization of the surrounding



(a) instance-wise discrimination (b) instance-level co-occurring semantic statistics (c) image-level co-occurring semantic statistics

Figure 5.2: We construct three types of contrastive relationships to encode local-to-global visual contexts in our learned feature mappings. Pixels are attracted to (repelled by) segments: **(a)** of the same (different) instance, **(b)** of similar (distinctive) semantic surrounds, and **(c)** belong to similar (different) image-level scenes. Such global scenes can be approximated by the occurrence of semantic categories in the image. The idea contextual feature mapper should capture both the visual appearance of the object itself, and the statistical distribution and spatial organization of the surrounds.

semantics. We optimize the pixel-wise feature mapper with three pixel-to-segment contrastive losses that encode local-to-global visual contexts: **1)** instance-wise discrimination, **2)** instance-level co-occurring semantic statistics, and **3)** image-level semantic co-occurrences (Fig. 5.2). We also introduce an additional regularization term, resulting in a total of 4 terms in the loss function.

Instance-wise discrimination. The idea is to push instances away from others such that only visually similar instances stay close in the feature space. Following [111], we contrast pixels with segments based on their instance labels C_O. Positive segments are the ones within pixel i's instance; negative segments include different-instance segments within and other than i's image.

Instance-level co-occurring semantic statistics. The surrounding context should also indicate how to develop feature mappings for each object instance. For example, a bike rider should be distinguished from a motorbike rider. We quantify the surrounding contexts and define contrastive relationship accordingly. Specifically, we calculate the semantic category distribution at the center and eight neighboring patches of the centered object, where the patch size is the same as object's height and width (Fig. 5.3). Within each patch, we measure the occurrence of each semantic category (including both *things* and *stuff*), resulting in a

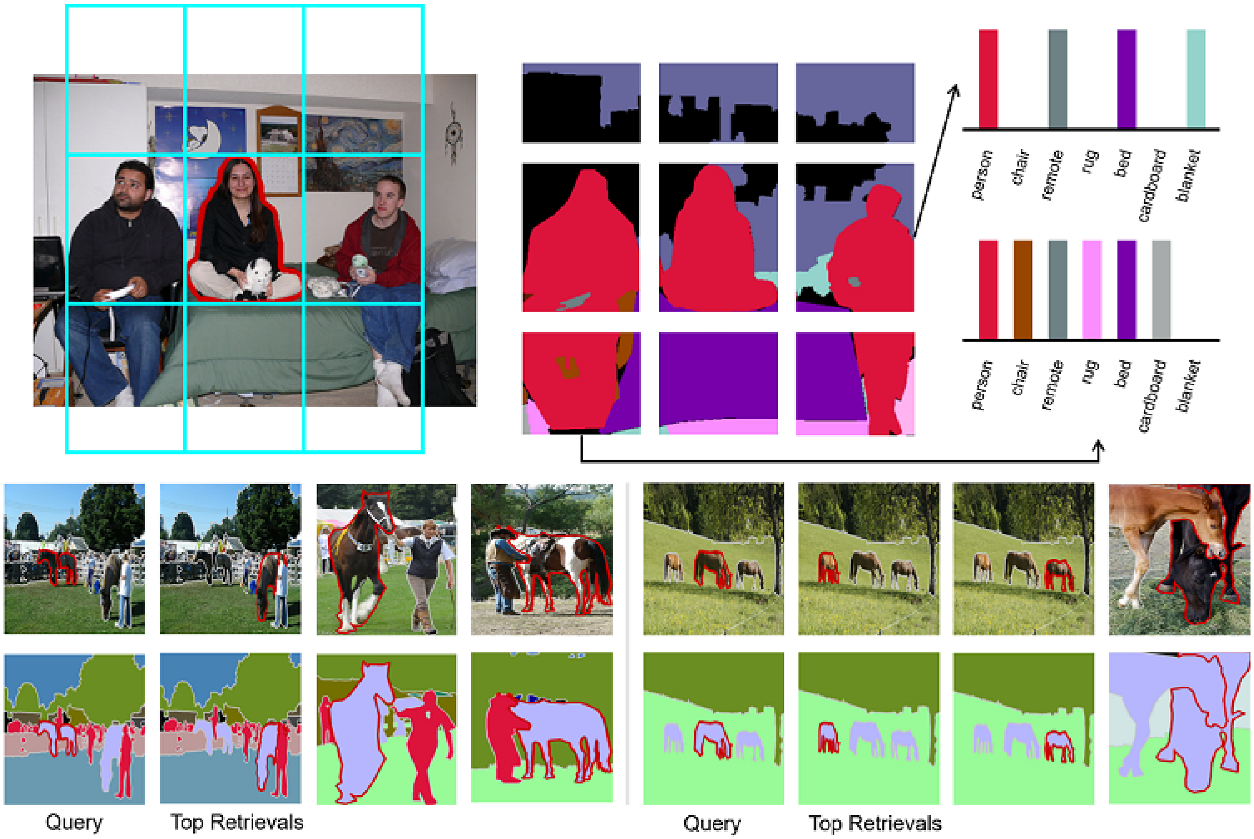


Figure 5.3: We quantify local visual contexts by calculating the semantic category distribution within each of the nine patches of the centered object (circled by red lines). The patch size is the same as object's height and width. **Top:** For each patch, we measure the occurrence of each semantic category, resulting in a binary contextual feature vector. **Bottom:** Based on such statistical contextual features, we conduct nearest neighbor search for each query object. Top-ranked retrievals have similar semantic surrounds as the query. We contrast pixels according to such second-order statistics to encode visual contexts.

binary contextual feature vector.

We define local context pseudo labels C_L based on such second-order statistics. We compute the Hamming distance between the contextual feature vectors of different objects. For pixel i, we define positive segments as the ones belonging to the top-ranked neighbors of i's object; others are negative segments. Positive segments are restricted to have the same semantic category. Our goal is to encourage groupings of object instances embedded in similar contexts.

Image-level co-occurring semantic statistics. We impose a more global regularization at the scene context level. Following [30], we characterize scene context in terms of the occurrences of semantic categories. Images with similar distribution of semantic categories tend to have similar scenes.

We ignore the spatial layout, and measure the occurrence of semantic categories within each image, from which we define global context pseudo labels C_G. For pixel i, its positive set includes all segments from the set of images which share at least one semantic category with i's image. All other segments are considered negative. That is, we desire pixels to be separated by their scene types.

Predictive coding regularization. We additionally impose a predictive coding loss to explicitly enforce structured correlation among pixel features. Intuitively, the feature at one pixel should help predict the feature at other pixels in the image. Following [231], we apply the regularization in a denoising autoencoding manner. We derive a noisy set of pixel features V' by randomly masking out a subset of pixel features from V. The goal is to reconstruct V from V' using multiple encoder and decoder layers. Let ψ be the autoencoder, the loss is: $L_M = \|V - \psi(V')\|_2^2$. See [231] for details.

Total training loss. There are 3 pixel-to-segment contrastive feature loss terms and 1 predictive coding regularization term: $L = \lambda_O L(C_O) + \lambda_L L(C_L) + \lambda_G L(C_G) + \lambda_M L_M$. The two types of losses are complementary to each other: The former enhances feature discrimination without any regard to spatial correlation, whereas the latter enforces structured correlation among pixels within an image, without any regard to instances in different images. We integrate these two aspects in the overall loss to optimize our contextual visual feature.

5.4 Experiments

We detail our training/testing procedures, and then benchmark our unsupervised visual context model on zero-shot recognition of human-object interactions and additionally semantic segmentation.

Datasets

HICO [189] is a generic human-object interaction dataset. It is labelled with 600 human-object interaction categories w.r.t 80 object categories. Both human and object bounding boxes are provided. The dataset has $38,118$ and $9,658$ images for training and testing.

MSCOCO [16] is a complex scene parsing dataset with 80 *things* and 91 *stuff* categories. Images have a high variety of visual scenes, such as dining, snow skiing, boat piloting, horse riding *etc.* . We adopt *train2017* split (118K images) for training.

Cityscapes [27] is a dataset for urban street scene parsing. It contains 19 *things* and *stuff* categories, such as road, pedestrian, and cars *etc.* . $5,000$ images are annotated with high-quality pixel labels, which are split into $2,975$, 500 and $1,525$ for training, validation and testing.

Pascal VOC 2012 is an object-centric semantic segmentation dataset, labelled with 20 object categories and a background class. Compared to MSCOCO, the image scenes are less complex, with an average of 2.3 objects occur per image (7.3 objects for MSCOCO).

We augment the training set with additional images [128], resulted in $10,582$ and $1,449$ for training and validation.

Architecture and Training

Architecture. For HOI recognition on HICO, we follow UPSNet [232] to build our model architecture. It consists of a ResNet50 [81] backbone, followed by a FPN [233] layer to generated multi-scale features. The channel dimension of output features are 256. We fuse the multi-scale features using a deformable convolutional [234] layer, resulted in 128-dim unit-length output features. For semantic segmentation on Cityscapes and VOC, we adopt deeplab-v2 [80] model architecture, where ResNet101 is used as the backbone CNN. The output feature dimension is set to 64.

Trainig. For all experiments, we fine-tune ResNet50 backbone, which is pre-trained on ImageNet [11] dataset. We use 2 Nvidia V100 cards for training. We set initial learning_rate to 0.003, momentum to 0.9, and weight_decay to 0.0001. Following [80], we adopt poly learning rate policy by multiplying base learning rate by $1 - (\frac{iter}{max\_iter})^{0.9}$.

On MSCOCO, we set crop_size to 640×640, batch_size to 12, training iterations to $60,000$. We iterate spherical K-Means algorithm for 10 steps to partition an image into 49 segments, which are furthered refined by instance and semantic pixel labels (see [48]). For contrastive losses, we set κ to 12, 16 and 16 for $L(C_O)$, $L(C_L)$ and $L(C_G)$. λ_O, λ_L, λ_G, λ_M are set to 1.0, 0.66, 0.5, and 1.0.

For training on Cityscapes and VOC, we set crop_size to 512×512, batch_size to 12, training iterations to $30,000$. We iterate spherical K-Means algorithm for 10 steps to partition an image into 36 segments. Such image oversegmentation is likewise refined by instance and semantic pixel labels. We adopt the same settings for the learning losses.

Supervised Baselines

We consider two kinds of supervised baseline methods for comparison: 1) Spatially Conditioned Graphs (SCG) [212], and 2) vanilla binary classifier. For SCG, we perform inference using HICO-trained ResNet50-FPN model weights, such that both object detector and interaction classifier are fine-tuned on HICO dataset. For vanilla binary classifier, we adopt exactly the same architecture as our method, but average pool pixel-wise features within each human bounding box. Additional two 1×1 convolutional layers are used as the binary classifier to predict the occurrence of each kind of interaction. Notably, SCG requires pairing a human with an object to classify their interaction, whereas, vanilla binary classify considers each human individually.

Oracle Baselines

We consider an oracle baseline method using ground-truth semantics for HOI recognition. On HICO dataset, we compute instance-level co-occurring semantic statistics using ground-truth bounding boxes. We convert bounding boxes into instance and semantic pixel labels. For each human instance, we calculate semantic category distribution at the center and eight neighboring patches. We perform nearest neighbor search using such binary context-induced features to infer HOI for the query human instance. Notably, our framework applies the procedure only during training on MSCOCO, whereas, the oracle baseline infers using HICO ground-truths.

Testing

For inference with our framework on HICO, we average pool and length normalize pixel-wise features within each human bounding box. We use the ground-truth human boxes but not object boxes. For SCG, object boxes are predicted with the object detector. For the oracle baseline and our method, We retrieve 20 nearest neighbors to predict interaction labels. For each query, we count the occurrence of each interaction category, and adjust the threshold from minimum to maximum number of occurrence. For both supervised baselines, we adjust the threshold w.r.t the classification scores. Threshold is applied to decide if the interaction category is detected. We plot the PR-cure and calculate AP performance correspondingly.

For inference of semantic segmentation on Cityscapes and VOC, we follow [48] to predict pixel labels by nearest neighbor search. See [48] for more details.

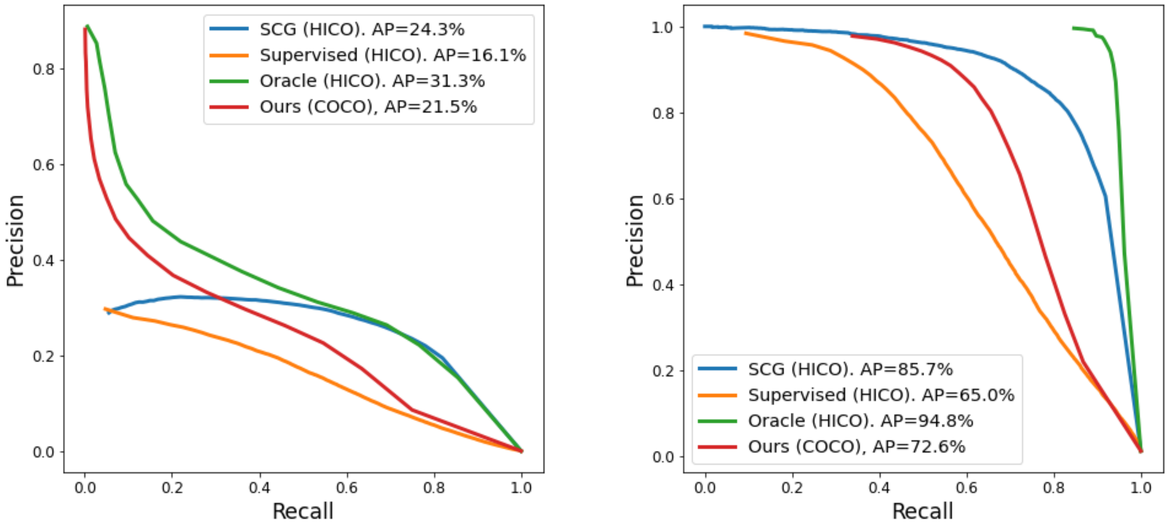
For all experiments, we dot not use multi-scale but only single-scale images during inference.

Quantitative Results on HOI recognition

We present the quantitative results for HOI recognition on HICO dataset. As shown in the left figure of Fig. 5.4, our method is upperbounded by the oracle baseline, and we achieve 68.7% of the oracle performance (21.5%$vs.$31.3% AP). Remarkably, our method has never seen any HICO image and label, but still obtains comparable performance w.r.t the SOTA supervised baseline: SCG (21.5%$vs.$24.3% AP). We report the performance based on the interacting object category not the interaction category in the right figure of Fig. 5.4. Our method achieves 76.5% and 84.7% performance with respect to the oracle and SCG baseline.

We summarize that training using ground-truth labels does not guarantee good testing performance to distinguish humans with different interactions. Our learned contextual features work as well as supervised classifiers. However, there is still room for improvement for our method to group instances according to co-occurring object semantics more precisely.

Note that HICO dataset annotates the human bounding boxes for each interaction label. Although the same human instance could have multiple interactions, we conduct inference on the human bounding box of each interaction label, individually. We do not filter out duplicated human instances, resulting in noisier predictions and less optimal performance than the ones reported in [212].



(a) Interaction category

(b) Interacting object category

Figure 5.4: Our unsupervised model approaches the supervised state-of-the-art for recognizing HOIs on HICO. **Left:** Performance evaluated on 600 interaction categories. **Right:** Performance evaluated on 80 interacting object categories. Our framework discovers high-level contextual relationships without any prior knowledge of relationship categories.

Dataset	Method	mIoU.
Cityscapes	SegSort Our framework	69.49 70.38
VOC	SegSort Our framework	75.98 77.71

Table 5.1: Our contextual regularizations improve semantic segmentation.

Quantitative Results on Semantic Segmentation

We summarize the efficacy of the proposed contextual regularizations for semantic segmentation on VOC and Cityscapes dataset in Table 5.1. Compared to SegSort [48], which uses only semantic pixel labels, we improve the semantic segmentation performance by 0.89% and 1.73% mIoU on Cityscapes and VOC. We show that our proposed regularizations help recognition in terms of capturing not only pixel itself, but also the surrounding contexts.

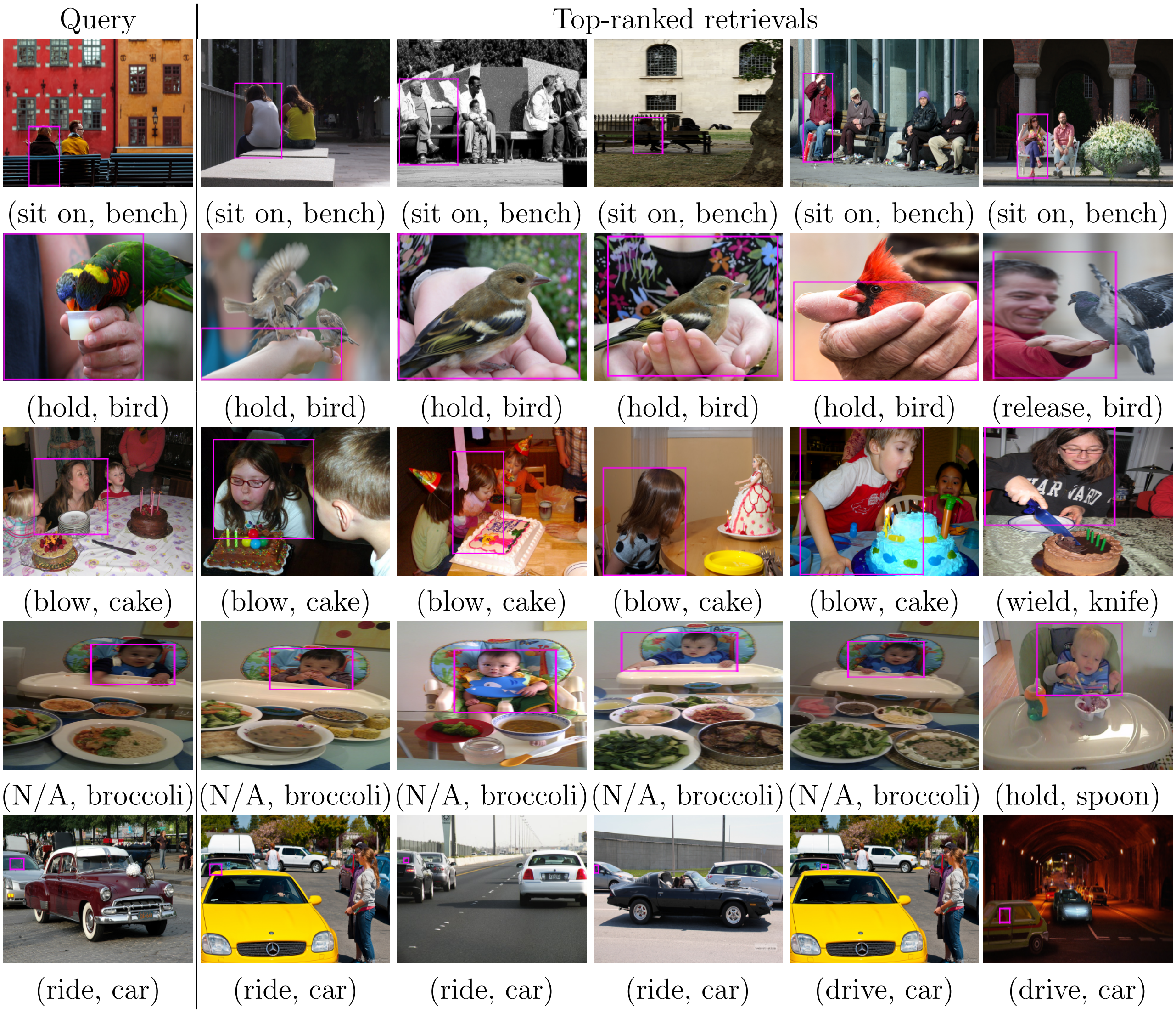


Figure 5.5: High-level contextual semantics emerge from our learned feature mappings. On HICO, we compute the average features within each human bounding box and conduct nearest neighbor retrievals. The ground-truth interaction label are shown in the form of (interaction, object) pair and put below each human instance. Strikingly, we found instances with the similar contextual relationships are close in the learned feature space. N/A denotes 'no interaction' category.

Visual results on Contextual Retrievals

We present visual results of nearest neighbor retrievals using our learned feature mappings in Fig. 5.5. Human instances of the similar contextual relationships are grouped.

$L(C_O)$	L_M	$L(C_G)$	$L(C_L)$	AP
✓	-	-	-	70.6
✓	✓	-	-	71.1
✓	✓	✓	-	71.9
✓	✓	✓	✓	72.6

λ_M	AP
0.0	71.7
0.5	72.3
1.0	72.6
2.0	71.1

λ_G	AP
0.0	71.6
0.5	72.6
1.0	71.6

λ_L	AP
0.0	70.9
0.66	72.6
1.0	71.6

Table 5.2: Each of our proposed loss regularizations helps feature mappings to capture visual contexts better. We report the performance evaluated by interacting object categories on HICO. **From left to right:** the performance gain resulted from the addition of each loss, and the effects of loss weightings.

Ablation Study on Proposed Regularizations

We summarize the efficacy of each proposed regularization in Table 5.2. We report the AP performance based on interacting object category. By successively adding loss terms L_M, $L(C_G)$ and $L(C_L)$, we improve the performance by 0.5%, 0.8% and 0.7% AP, compared to the models training with only instance discrimination loss. We also study the weightings for each loss, and adopt the best set of hyper-parameters for training.

5.5 Summary

We develop a contextual visual feature learning model to tackle recognition of human-object interactions. Without any supervision on relationships, our model approaches the supervised state-of-the-art and is able to discover such high-level contextual relationships automatically.

Chapter 6

Discussion

In this dissertation, we extensively study the recognition problem in four different aspects given minimal human-labeled supervision. Now, we summarize and highlight each of their contributions.

We first study the weakly-supervised semantic segmentation problem and propose a unified framework to tackle all types of weak annotations in Chapter 2. Our insight is to formulate weakly supervised segmentation as a semi-supervised metric learning problem, where we supervise features of both labeled and unlabeled pixels with partial annotations. In particular, we propose four kinds of contrastive relationships: low-level image similarity, semantic annotations, semantic co-occurrence, and feature affinity. Using these relationships, we involve both labeled and unlabeled pixels in discriminative feature learning. Compared to other alternatives, we can tackle all types of annotations consistently. On Pascal VOC, we demonstrate superior performance on each type of weak annotation. On DensePose, training with point annotations, our results outperform baselines by a large margin.

We next study the problem of unsupervised semantic segmentation, which is more challenging than its weakly-supervised counterpart. Without any labeled supervision, it is impossible to name the semantic categories of image pixels. The task is to group, not classify, pixels in unlabeled images. We consider unsupervised segmentation as a pixel-wise feature learning problem. However, semantics have different levels of granularity, and existing methods ignore the ambiguity of granularity. Instead, we embrace it and exploit the hierarchical structure to develop our pixel features. Our contribution has two folds: **1)** we enforce spatial consistency of groupings, such that features of the same region shall be invariant to visual changes, and **2)** we enforce hierarchical consistency of groupings, such that features shall be consistent with fine-to-coarse grouping cues. Our unsupervised semantic segmentation achieved SOTA on both object- and scene-centric benchmark datasets (Pascal VOC, Cityscapes, KITTI-STEP, Potsdam, COCO-Stuff). On Pascal VOC, our hierarchical segmentation outperforms other clustering methods by a large margin. Lastly, we showcase contextual retrievals on COCO across different levels of granularity.

We next propose to unify image classification and segmentation in a single framework. Existing methods require two separate models to tackle each of the tasks. Classification and segmentation results are thus independent of each other. Instead, we consider them as concurrent tasks. Building upon ViT architectures, our model has three novel aspects. **1)** we use segment, not patch, tokens. **2)** we create a token hierarchy with the proposed graph pooling module, which results in a hierarchical segmentation of the image. **3)** our model does not require human-labeled supervision. For classification on ImageNet, our model achieves better tradeoffs between model efficiency and task performance. For segmentation on Pascal VOC, we demonstrate better performance with less computation. Notably, our model does not require additional pixel clustering or CRF post-processing, and our segmentations are still better at capturing object boundaries. Lastly, our attention-induced figure-ground segmentations outperform DINO by a large margin.

Lastly, we tackle unsupervised human-object relationship recognition. Existing methods require supervision of relationships, which becomes impractical as the number of object categories increases and the number of human-object relationships exponentially explodes. We address the issue by transforming the relationship classification problem into a discriminative feature learning problem: image pixels in similar relationships are mapped to similar points in the latent feature space and vice versa. In particular, we do not use relationship labels for training but enforce groupings of pixel features based on similarities of visual contexts. Our key insight is that objects surrounded by similar contexts have similar relationships. We characterize the visual context of pixels with their surrounding semantic and spatial configuration. We contrast pixel features to capture: **1)** instance ownerships, **2)** semantic co-occurrence statistics, and **3)** structure correlations. On HICO dataset, our unsupervised method achieves competitive results with supervised baselines.

To summarize, we explore general feature learning and grouping models which need minimal human-labeled supervision. Our works suggest robust approaches to understanding the highly-complex and fast-changing real-world scenes.

Bibliography

[1] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in Neural Information Processing Systems* 25 (2012).

[3] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[4] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[5] Roberto Brunelli and Tomaso Poggio. “Face recognition: Features versus templates”. In: *IEEE transactions on pattern analysis and machine intelligence* 15.10 (1993), pp. 1042–1052.

[6] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 711–720.

[7] Gabriella Csurka et al. “Visual categorization with bags of keypoints”. In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. Prague. 2004, pp. 1–2.

[8] Jorge Sánchez et al. “Image classification with the fisher vector: Theory and practice”. In: *International journal of computer vision* 105.3 (2013), pp. 222–245.

[9] Robert Fergus, Pietro Perona, and Andrew Zisserman. “Object class recognition by unsupervised scale-invariant learning”. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. IEEE. 2003, pp. II–II.

[10] Li Fei-Fei, Robert Fergus, and Pietro Perona. “One-shot learning of object categories”. In: *IEEE transactions on pattern analysis and machine intelligence* 28.4 (2006), pp. 594–611.

[11] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* Ieee. 2009, pp. 248–255.

[12] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[13] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587.

[14] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2010), pp. 1627–1645.

[15] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *International Journal of Computer Vision* (2010).

[16] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014, pp. 740–755.

[17] Alexander Kirillov et al. “Panoptic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9404–9413.

[18] D. Martin et al. “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Vol. 2. 2001, pp. 416–423.

[19] Yu-Wei Chao et al. “Hico: A benchmark for recognizing human-object interactions in images”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015, pp. 1017–1025.

[20] Saurabh Gupta and Jitendra Malik. “Visual semantic role labeling”. In: *arXiv preprint arXiv:1505.04474* (2015).

[21] Andrew Rabinovich et al. “Objects in Context.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2007.

[22] Yong Jae Lee and Kristen Grauman. “Object-graphs for context-aware category discovery”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 1–8.

[23] Kaihua Tang et al. “Learning to compose dynamic tree structures for visual contexts”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6619–6628.

[24] Tomasz Malisiewicz and Alyosha Efros. “Beyond categories: The visual memex model for reasoning about object relationships”. In: *NIPS*. 2009.

[25] Georgia Gkioxari et al. “Detecting and recognizing human-object interactions”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8359–8367.

[26] Siyuan Qi et al. “Learning human-object interactions by graph parsing neural networks”. In: *European Conference on Computer Vision*. 2018, pp. 401–417.

[27] Marius Cordts et al. "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3213–3223.

[28] Ranjay Krishna et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International journal of computer vision* 123.1 (2017), pp. 32–73.

[29] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. "Densepose: Dense human pose estimation in the wild". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7297–7306.

[30] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. "Universal Weakly Supervised Segmentation by Pixel-to-Segment Contrastive Learning". In: *International Conference on Learning Representations*. 2021.

[31] Tsung-Wei Ke et al. "Unsupervised Hierarchical Semantic Segmentation with Multiview Cosegmentation and Clustering Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2571–2581.

[32] Tsung-Wei Ke and Stella X Yu. "CAST: Concurrent Recognition and Segmentation with Adaptive Segment Tokens". In: *arXiv preprint arXiv:2210.00314* (2022).

[33] Tsung-Wei Ke et al. "Contextual Visual Feature Learning for Zero-Shot Recognition of Human-Object Interactions". In: *Advances in Neural Information Processing Systems Workshop on Human in the Loop Learning* (2022).

[34] Yu-Ting Chang et al. "Weakly-Supervised Semantic Segmentation via Sub-category Exploration". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020).

[35] Chunfeng Song et al. "Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3136–3145.

[36] Meng Tang et al. "On regularized losses for weakly-supervised cnn segmentation". In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 507–522.

[37] Bolei Zhou et al. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2921–2929.

[38] Alexander Kolesnikov and Christoph H Lampert. "Seed, expand and constrain: Three principles for weakly-supervised image segmentation". In: *Proceedings of the European Conference on Computer Vision*. Springer. 2016, pp. 695–711.

[39] Jiwoon Ahn and Suha Kwak. "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4981–4990.

[40] Zilong Huang et al. "Weakly-supervised semantic segmentation network with deep seeded region growing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7014–7023.

[41] Jungbeom Lee et al. "Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5267–5276.

[42] Jifeng Dai, Kaiming He, and Jian Sun. "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015, pp. 1635–1643.

[43] Anna Khoreva et al. "Simple does it: Weakly supervised instance and semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2017, pp. 876–885.

[44] Amy Bearman et al. "What's the point: Semantic segmentation with point supervision". In: *Proceedings of the European Conference on Computer Vision*. Springer. 2016, pp. 549–565.

[45] Di Lin et al. "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3159–3167.

[46] Meng Tang et al. "Normalized cut loss for weakly-supervised cnn segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1818–1827.

[47] Philipp Krähenbühl and Vladlen Koltun. "Efficient inference in fully connected crfs with gaussian edge potentials". In: *Advances in Neural Information Processing Systems*. 2011, pp. 109–117.

[48] Jyh-Jing Hwang et al. "SegSort: Segmentation by Discriminative Sorting of Segments". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

[49] Jason Weston et al. "Deep learning via semi-supervised embedding". In: *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 639–655.

[50] Durk P Kingma et al. "Semi-supervised learning with deep generative models". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3581–3589.

[51] Antti Rasmus et al. "Semi-supervised learning with ladder networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 3546–3554.

[52] Takeru Miyato et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.8 (2018), pp. 1979–1993.

[53] Antti Tarvainen and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1195–1204.

[54] Kevin Clark, Thang Luong, and Quoc V Le. “Cross-view training for semi-supervised learning”. In: *arXiv preprint arXiv:1809.08370* (2018).

[55] Thorsten Joachims. “Transductive learning via spectral graph partitioning”. In: *International Conference on Machine Learning*. 2003, pp. 290–297.

[56] Dengyong Zhou et al. “Learning with local and global consistency”. In: *Advances in Neural Information Processing Systems*. 2004, pp. 321–328.

[57] Rob Fergus, Yair Weiss, and Antonio Torralba. “Semi-supervised learning in gigantic image collections”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 522–530.

[58] Bin Liu et al. “Deep metric transfer for label propagation with limited annotated data”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019.

[59] Bin Wang et al. “Boundary Perception Guidance: A Scribble-Supervised Semantic Segmentation Approach.” In: *International Joint Conference on Artificial Intelligence* (2019).

[60] George Papandreou et al. “Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015, pp. 1742–1750.

[61] Kunpeng Li et al. “Tell me where to look: Guided attention inference network”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9215–9223.

[62] Wataru Shimoda and Keiji Yanai. “Self-supervised difference detection for weakly-supervised semantic segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5208–5217.

[63] Bingfeng Zhang et al. “Reliability Does Matter: An End-to-End Weakly Supervised Semantic Segmentation Approach”. In: *arXiv preprint arXiv:1911.08039* (2019).

[64] Qi Yao and Xiaojin Gong. “Saliency Guided Self-Attention Network for Weakly and Semi-Supervised Semantic Segmentation”. In: *IEEE Access* 8 (2020), pp. 14413–14423.

[65] Nikita Araslanov and Stefan Roth. “Single-Stage Semantic Segmentation from Image Labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020).

[66] Yude Wang et al. “Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2020).

[67] Junsong Fan et al. “Learning Integral Objects With Intra-Class Discriminator for Weakly-Supervised Semantic Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[68] Guolei Sun et al. “Mining Cross-Image Semantics for Weakly Supervised Semantic Segmentation”. In: *Proceedings of the European Conference on Computer Vision*. 2020.

[69] Jia Xu, Alexander G Schwing, and Raquel Urtasun. “Learning to segment under various forms of weak supervision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3781–3790.

[70] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. “Constrained convolutional neural networks for weakly supervised segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015, pp. 1796–1804.

[71] Bryan Russell et al. “Segmenting scenes by matching image composites”. In: *Advances in Neural Information Processing Systems*. 2009.

[72] Joseph Tighe and Svetlana Lazebnik. “Superparsing: scalable nonparametric image parsing with superpixels”. In: *Proceedings of the European Conference on Computer Vision*. 2010.

[73] Ce Liu, Jenny Yuen, and Antonio Torralba. “Nonparametric scene parsing via label transfer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2011).

[74] Zhirong Wu et al. “Unsupervised feature learning via non-parametric instance-level discrimination”. In: *arXiv preprint arXiv:1805.01978* (2018).

[75] Zhirong Wu, Alexei A Efros, and Stella X Yu. “Improving generalization via scalable neighborhood component analysis”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 685–701.

[76] Saining Xie and Zhuowen Tu. “Holistically-nested edge detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015.

[77] Pablo Arbelaez et al. “Contour detection and hierarchical image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2010), pp. 898–916.

[78] Arindam Banerjee et al. “Clustering on the unit hypersphere using von Mises-Fisher distributions”. In: *Journal of Machine Learning Research* 6.Sep (2005), pp. 1345–1382.

[79] Jacob Goldberger et al. “Neighbourhood components analysis”. In: *Advances in Neural Information Processing Systems* 17 (2004).

[80] Liang-Chieh Chen et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[81] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[82] Hengshuang Zhao et al. “Pyramid scene parsing network”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2881–2890.

[83] John Towns et al. “XSEDE: accelerating scientific discovery”. In: *Computing in Science & Engineering* 16.5 (2014), pp. 62–74.

[84] Nicholas A Nystrom et al. “Bridges: a uniquely flexible HPC resource for new communities and data analytics”. In: *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*. 2015, pp. 1–8.

[85] Wouter Van Gansbeke et al. “Revisiting Contrastive Methods for Unsupervised Learning of Visual Representations”. In: *arXiv preprint arXiv:2106.05967* (2021).

[86] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9729–9738.

[87] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1597–1607.

[88] Xudong Wang, Ziwei Liu, and Stella X Yu. “Unsupervised Feature Learning by Cross-Level Instance-Group Discrimination”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12586–12595.

[89] 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*. 2019, pp. 9865–9874.

[90] Xinlong Wang et al. “Dense contrastive learning for self-supervised visual pre-training”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3024–3033.

[91] Stella X. Yu. “Segmentation Induced by Scale Invariance”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2005.

[92] Dorin Comaniciu and Peter Meer. “Mean shift: A robust approach toward feature space analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002).

[93] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Efficient graph-based image segmentation”. In: *International Journal of Computer Vision* (2004).

[94] Jianbo Shi and Jitendra Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905.

[95] Jitendra Malik et al. “Contour and texture analysis for image segmentation”. In: *International Journal of Computer Vision* (2001).

[96] Stella X. Yu and Jianbo Shi. “Multiclass spectral clustering”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2003.

[97] Stella X Yu and Jianbo Shi. “Segmentation given partial grouping constraints”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2004).

[98] Jyh-Jing Hwang and Tyng-Luh Liu. “Pixel-wise deep learning for contour detection”. In: *arXiv preprint arXiv:1504.01989* (2015).

[99] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551.

[100] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[101] Liang-Chieh Chen et al. “Rethinking atrous convolution for semantic image segmentation”. In: *arXiv preprint arXiv:1706.05587* (2017).

[102] Xuming He, Richard S Zemel, and MA Carreira-Perpindn. “Multiscale conditional random fields for image labeling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2004.

[103] Jamie Shotton et al. “Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context”. In: *International Journal of Computer Vision* (2009).

[104] Pushmeet Kohli, Philip HS Torr, et al. “Robust higher order potentials for enforcing label consistency”. In: *International Journal of Computer Vision* 82.3 (2009), pp. 302–324.

[105] Lubor Ladicky et al. “Associative hierarchical crfs for object class image segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2009.

[106] Stephen Gould, Richard Fulton, and Daphne Koller. “Decomposing a scene into geometric and semantically consistent regions”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2009.

[107] Jian Yao, Sanja Fidler, and Raquel Urtasun. “Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2012.

[108] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. “Feedforward semantic segmentation with zoom-out features”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2015.

[109] Tsung-Wei Ke et al. “Adaptive Affinity Fields for Semantic Segmentation”. In: *Proceedings of the European Conference on Computer Vision*. 2018.

[110] Jyh-Jing Hwang et al. “Adversarial Structure Matching for Structured Prediction Tasks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[111] Jyh-Jing Hwang, Tsung-Wei Ke, and Stella X Yu. “Contextual Image Parsing via Panoptic Segment Sorting”. In: *Multimedia Understanding with Less Labeling on Multimedia Understanding with Less Labeling*. 2021, pp. 27–36.

[112] Ramprasaath R Selvaraju et al. “Casting your model: Learning to localize improves self-supervised representations”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11058–11067.

[113] Xiao Zhang and Michael Maire. “Self-Supervised Visual Representation Learning from Hierarchical Grouping”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[114] Claude Elwood Shannon. “A mathematical theory of communication”. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[115] Kuan-Yun Lee. “New Information Inequalities with Applications to Statistics”. PhD thesis. EECS Department, University of California, Berkeley, May 2022. URL: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-37.html`.

[116] Yassine Ouali, Celine Hudelot, and Myriam Tami. “Autoregressive Unsupervised Image Segmentation”. In: *Proceedings of the European Conference on Computer Vision*. 2020.

[117] Zhenli Zhang et al. “ExFuse: Enhancing Feature Fusion for Semantic Segmentation”. In: *Proceedings of the European Conference on Computer Vision*. 2018.

[118] Carsten Rother et al. “Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Vol. 1. IEEE. 2006, pp. 993–1000.

[119] Armand Joulin, Francis Bach, and Jean Ponce. “Discriminative clustering for image co-segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 1943–1950.

[120] Yong Jae Lee and Kristen Grauman. “Collect-cut: Segmentation with top-down cues discovered in multi-object images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 3185–3192.

[121] Marina Meila and Jianbo Shi. “Learning Segmentation by Random Walks”. In: *Advances in Neural Information Processing Systems*. 2000.

[122] Piotr Dollár and C Lawrence Zitnick. “Fast edge detection using structured forests”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.8 (2014), pp. 1558–1570.

[123] Nicolas Carion et al. “End-to-end object detection with transformers”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 213–229.

[124] Ulrike Von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and Computing* 17.4 (2007), pp. 395–416.

[125] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. “Efficient parameter-free clustering using first neighbor relations”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8934–8943.

[126] Anton Tsitsulin et al. “Graph clustering with graph neural networks”. In: *arXiv preprint arXiv:2006.16904* (2020).

[127] Mark EJ Newman. “Finding community structure in networks using the eigenvectors of matrices”. In: *Physical Review E* 74.3 (2006), p. 036104.

[128] Bharath Hariharan et al. “Semantic contours from inverse detectors”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE. 2011, pp. 991–998.

[129] Mark Weber et al. “STEP: Segmenting and Tracking Every Pixel”. In: *arXiv preprint arXiv:2102.11859* (2021).

[130] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. “Coco-stuff: Thing and stuff classes in context”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1209–1218.

[131] Markus Gerke. “Use of the stair vision library within the ISPRS 2D semantic labeling benchmark”. In: (2014).

[132] Phillip Isola et al. “Crisp boundary detection using pointwise mutual information”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2014, pp. 799–814.

[133] Mathilde Caron et al. “Deep clustering for unsupervised learning of visual features”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 132–149.

[134] Carl Doersch, Abhinav Gupta, and Alexei A Efros. “Unsupervised visual representation learning by context prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015, pp. 1422–1430.

[135] Phillip Isola et al. “Learning visual groups from co-occurrences in space and time”. In: *arXiv preprint arXiv:1511.06811* (2015).

[136] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* (2008).

[137] Andrew P Witkin and Jay M Tenenbaum. “On the role of structure in vision”. In: *Human and Machine Vision*. Elsevier, 1983, pp. 481–543.

[138] Irving Biederman. “Recognition-by-components: a theory of human image understanding.” In: *Psychological Review* 94.2 (1987), p. 115.

[139] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[140] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[141] Wenjie Luo et al. “Understanding the effective receptive field in deep convolutional neural networks”. In: *Advances in Neural Information Processing Systems* 29 (2016).

[142] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[143] Saurabh Goyal et al. "PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3690–3699.

[144] Dmitrii Marin et al. "Token pooling in vision transformers". In: *arXiv preprint arXiv:2110.03860* (2021).

[145] Xinlei Chen, Saining Xie, and Kaiming He. "An empirical study of training self-supervised vision transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9640–9649.

[146] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10347–10357.

[147] Li Yuan et al. "Tokens-to-token vit: Training vision transformers from scratch on imagenet". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 558–567.

[148] Wenhai Wang et al. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 568–578.

[149] Niki Parmar et al. "Image transformer". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4055–4064.

[150] Prajit Ramachandran et al. "Stand-alone self-attention in vision models". In: *Advances in Neural Information Processing Systems* 32 (2019).

[151] Iz Beltagy, Matthew E Peters, and Arman Cohan. "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150* (2020).

[152] Rewon Child et al. "Generating long sequences with sparse transformers". In: *arXiv preprint arXiv:1904.10509* (2019).

[153] Manzil Zaheer et al. "Big bird: Transformers for longer sequences". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17283–17297.

[154] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer". In: *arXiv preprint arXiv:2001.04451* (2020).

[155] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. "Fast transformers with clustered attention". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21665–21674.

[156] Yi Tay et al. "Sparse sinkhorn attention". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9438–9447.

[157] Peter J Liu et al. "Generating wikipedia by summarizing long sequences". In: *arXiv preprint arXiv:1801.10198* (2018).

[158] Sinong Wang et al. "Linformer: Self-attention with linear complexity". In: *arXiv preprint arXiv:2006.04768* (2020).

[159] Byeongho Heo et al. "Rethinking spatial dimensions of vision transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2021, pp. 11936–11945.

[160] Xiaoyi Dong et al. "Cswin transformer: A general vision transformer backbone with cross-shaped windows". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2022, pp. 12124–12134.

[161] Yongming Rao et al. "Dynamicvit: Efficient vision transformers with dynamic token sparsification". In: *Advances in Neural Information Processing Systems* 34 (2021).

[162] Wang Zeng et al. "Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering Transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2022, pp. 11101–11111.

[163] Jiarui Xu et al. "GroupViT: Semantic Segmentation Emerges from Text Supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2022, pp. 18134–18144.

[164] Xiaofeng Ren and Jitendra Malik. "Learning a classification model for segmentation". In: *Computer Vision, IEEE International Conference on.* Vol. 2. IEEE Computer Society. 2003, pp. 10–10.

[165] Greg Mori et al. "Recovering human body configurations: Combining segmentation and recognition". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 2. IEEE. 2004, pp. II–II.

[166] Zhixiang Ren et al. "Region-based saliency detection and its application in object recognition". In: *IEEE Transactions on Circuits and Systems for Video Technology* 24.5 (2013), pp. 769–779.

[167] Stephen Gould et al. "Multi-class segmentation with relative location prior". In: *International journal of computer vision* 80.3 (2008), pp. 300–316.

[168] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. "Class segmentation and object localization with superpixel neighborhoods". In: *2009 IEEE 12th international conference on computer vision.* IEEE. 2009, pp. 670–677.

[169] Abhishek Sharma, Oncel Tuzel, and Ming-Yu Liu. "Recursive context propagation network for semantic scene labeling". In: *Advances in Neural Information Processing Systems* 27 (2014).

[170] Raghudeep Gadde et al. "Superpixel convolutional networks using bilateral inceptions". In: *European Conference on Computer Vision.* 2016.

[171] Xing Wei et al. “Superpixel hierarchy”. In: *IEEE Transactions on Image Processing* 27.10 (2018), pp. 4838–4849.

[172] Yifan Zhang, Bo Pang, and Cewu Lu. “Semantic Segmentation by Early Region Proxy”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1258–1268.

[173] Stella X. Yu, Ralph Gross, and Jianbo Shi. “Concurrent object recognition and segmentation by graph partitioning”. In: *Advances in neural information processing systems* 15 (2002).

[174] Stella X. Yu and Jianbo Shi. “Object-specific figure-ground segregation”. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. IEEE. 2003, pp. II–39.

[175] Michael Maire. “Simultaneous segmentation and figure/ground organization using angular embedding”. In: *European Conference on Computer Vision*. Springer. 2010, pp. 450–464.

[176] Michael Maire, Stella X. Yu, and Pietro Perona. “Object detection and segmentation from joint embedding of parts and pixels”. In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 2142–2149.

[177] Michael Van den Bergh et al. “Seeds: Superpixels extracted via energy-driven sampling”. In: *European Conference on Computer Vision*. Springer. 2012, pp. 13–26.

[178] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[179] Charles Ruizhongtai Qi et al. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[180] Bharath Hariharan et al. “Hypercolumns for object segmentation and fine-grained localization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2015, pp. 447–456.

[181] Yonglong Tian, Dilip Krishnan, and Phillip Isola. “Contrastive multiview coding”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 776–794.

[182] Tete Xiao et al. “Early convolutions help transformers see better”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[183] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[184] Mathilde Caron et al. “Emerging properties in self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9650–9660.

[185] Aude Oliva and Antonio Torralba. “The role of context in object recognition”. In: *Trends in cognitive sciences* 11.12 (2007), pp. 520–527.

[186] Myung Jin Choi et al. “Exploiting hierarchical context on a large database of object categories”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 129–136.

[187] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. “A tree-based context model for object recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.2 (2011), pp. 240–252.

[188] Roozbeh Mottaghi et al. “The role of context for object detection and semantic segmentation in the wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2014, pp. 891–898.

[189] Yu-Wei Chao et al. “Learning to detect human-object interactions”. In: *2018 ieee winter conference on applications of computer vision (wacv)*. IEEE. 2018, pp. 381–389.

[190] Alina Kuznetsova et al. “The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale”. In: *IJCV* (2020).

[191] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. “UCF101: A dataset of 101 human actions classes from videos in the wild”. In: *arXiv preprint arXiv:1212.0402* (2012).

[192] Chunhui Gu et al. “Ava: A video dataset of spatio-temporally localized atomic visual actions”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6047–6056.

[193] Bryan A Plummer et al. “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015, pp. 2641–2649.

[194] Stanislaw Antol et al. “Vqa: Visual question answering”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2015, pp. 2425–2433.

[195] Drew A Hudson and Christopher D Manning. “Gqa: A new dataset for real-world visual reasoning and compositional question answering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6700–6709.

[196] Qingqiu Huang et al. “Movienet: A holistic dataset for movie understanding”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 709–727.

[197] Xinlei Chen et al. “Iterative visual reasoning beyond convolutions”. In: *PProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7239–7248.

[198] Xinlei Chen and Abhinav Gupta. “Spatial memory for context reasoning in object detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2017.

[199] Bohan Zhuang et al. “Hcvrd: a benchmark for large-scale human-centered visual relationship detection”. In: *AAAI Conference on Artificial Intelligence*. 2018.

[200] Tanmay Gupta, Alexander Schwing, and Derek Hoiem. “No-frills human-object interaction detection: Factorization, layout encodings, and training techniques”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9677–9685.

[201] Tiancai Wang et al. “Learning human-object interaction detection using interaction points”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4116–4125.

[202] Chen Gao, Yuliang Zou, and Jia-Bin Huang. “ican: Instance-centric attention network for human-object interaction detection”. In: *arXiv preprint arXiv:1808.10437* (2018).

[203] Bangpeng Yao and Li Fei-Fei. “Modeling mutual context of object and human pose in human-object interaction activities”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 17–24.

[204] Yue Liao et al. “Ppdm: Parallel point detection and matching for real-time human-object interaction detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 482–490.

[205] Tiancai Wang et al. “Deep contextual attention for human-object interaction detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5694–5702.

[206] Dong-Jin Kim et al. “Detecting human-object interactions with action co-occurrence priors”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 718–736.

[207] Cheng Zou et al. “End-to-end human object interaction detection with hoi transformer”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11825–11834.

[208] Qi Dong et al. “Visual relationship detection using part-and-sum transformers with composite queries”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3550–3559.

[209] Aixi Zhang et al. “Mining the Benefits of Two-stage and One-stage HOI Detection”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[210] Mingfei Chen et al. “Reformulating hoi detection as adaptive set prediction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9004–9013.

[211] Chen Gao et al. “DRG: Dual Relation Graph for Human-Object Interaction Detection”. In: *European Conference on Computer Vision*. 2020.

[212] Frederic Z Zhang, Dylan Campbell, and Stephen Gould. “Spatially conditioned graphs for detecting human-object interactions”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13319–13327.

[213] Hai Wang, Wei-shi Zheng, and Ling Yingbiao. “Contextual heterogeneous graph network for human-object interaction detection”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 248–264.

[214] Oytun Ulutan, ASM Iftekhar, and Bangalore S Manjunath. “Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13617–13626.

[215] Yong-Lu Li et al. “Detailed 2D-3D Joint Representation for Human-Object Interaction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[216] Yong-Lu Li et al. “Transferable Interactiveness Knowledge for Human-Object Interaction Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[217] Ankan Bansal et al. “Detecting Human-Object Interactions via Functional Generalization.” In: *AAAI Conference on Artificial Intelligence*. 2020.

[218] Keizo Kato, Yin Li, and Abhinav Gupta. “Compositional learning for human object interaction”. In: *European Conference on Computer Vision*. 2018.

[219] Julia Peyre et al. “Detecting unseen visual relations using analogies”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1981–1990.

[220] Bingjie Xu et al. “Learning to Detect Human-Object Interactions With Knowledge”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[221] Bo Wan et al. “Pose-aware multi-level feature network for human object interaction detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9469–9478.

[222] Bingjie Xu et al. “Interact as You Intend: Intention-Driven Human-Object Interaction Detection”. In: *IEEE Transactions on Multimedia* (2019).

[223] Julia Peyre et al. “Weakly-supervised learning of visual relations”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2017.

[224] Hao-Shu Fang et al. “Pairwise body-part attention for recognizing human-object interactions”. In: *European Conference on Computer Vision*. 2018, pp. 51–67.

[225] Zhi Hou et al. “Visual Compositional Learning for Human-Object Interaction Detection”. In: *European Conference on Computer Vision*. 2020.

[226] Kongming Liang et al. “Visual relationship detection with deep structural ranking”. In: *AAAI Conference on Artificial Intelligence*. 2018.

[227] Suchen Wang et al. “Discovering human interactions with novel objects via zero-shot learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11652–11661.

[228] Dong-Jin Kim et al. "ACP++: Action Co-Occurrence Priors for Human-Object Interaction Detection". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 9150–9163.

[229] Gerard Salton and Michael J McGill. *Introduction to modern information retrieval.* mcgraw-hill, 1983.

[230] Christian Buchta et al. "Spherical k-means clustering". In: *Journal of Statistical Software* (2012).

[231] Kaiming He et al. "Masked autoencoders are scalable vision learners". In: *arXiv preprint arXiv:2111.06377* (2021).

[232] Yuwen Xiong et al. "Upsnet: A unified panoptic segmentation network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2019, pp. 8818–8826.

[233] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2017, pp. 2117–2125.

[234] Jifeng Dai et al. "Deformable convolutional networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2017, pp. 764–773.