CAST: CONCURRENT RECOGNITION AND SEGMENTATION WITH ADAPTIVE SEGMENT TOKENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recognizing an image and segmenting it into coherent regions are often treated as separate tasks. Human vision, however, has a general sense of segmentation hierarchy before recognition occurs. We are thus inspired to learn image recognition with hierarchical image segmentation based entirely on unlabeled images. Our insight is to learn fine-to-coarse features concurrently at superpixels, segments, and full image levels, enforcing consistency and goodness of feature induced segmentations while maximizing discrimination among image instances.

Our model innovates vision transformers on three aspects. **1)** We use adaptive segment tokens instead of fixed-shape patch tokens. **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 produce hierarchical image segmentation for free *while* training for recognition by maximizing image-wise discrimination.

Our work delivers the first concurrent recognition and hierarchical segmentation model *without any supervision*. Validated on ImageNet and PASCAL VOC, it achieves better recognition and segmentation with higher computational efficiency.

1 INTRODUCTION

Convolutional neural networks (CNN) (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016) and Vision Transformers (ViT) (Dosovitskiy et al., 2020) 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 (Martin et al., 2001). Fig. 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 (Hwang et al., 2019; Ke et al., 2022).

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 (Witkin & Tenenbaum, 1983; Biederman, 1987) has been overlooked in CNN and ViT architectures. Models optimized for image classification have no concepts of parts and wholes; recognition is achieved without understanding how different parts such as *eyes* and *face* are organized for the whole *animal*.

To understand the connections between parts and wholes, visual information must be extracted locally and globally. There are three major approaches (Fig. 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 (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014), uniform sub-sampling (striding) (He et al., 2016) and patch merging (Liu et al., 2021) are performed multiple times to increase the effective receptive field size (Luo et al., 2016). To generate a segmentation, the output features need to be upsampled and clustered (e.g. K-Means (Hwang et al., 2019)), often resulting in over-smoothed boundaries.
2. **Attention:** Inspired by Natural Language Processing (NLP), image patches are treated 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 (Vaswani et al., 2017). However, ViT is computationally inefficient as all image tokens are kept in every transformer block.

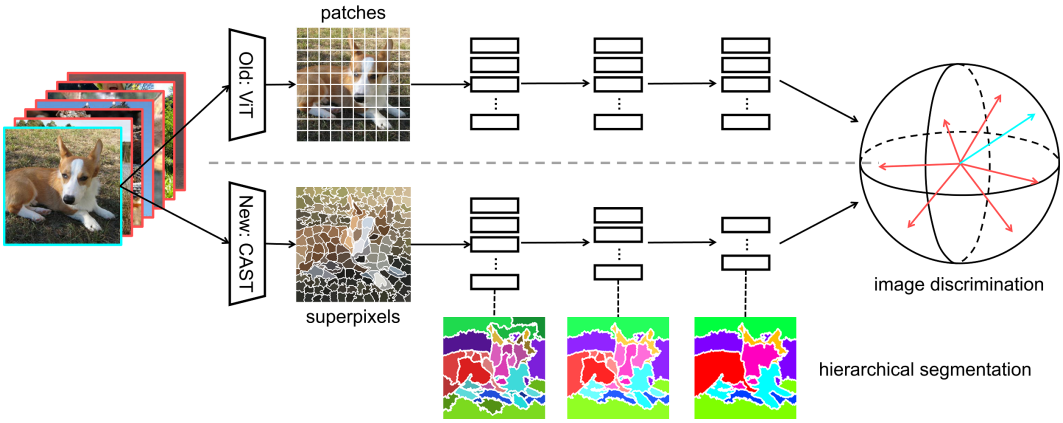


Figure 1: We innovate vision transformer models to concurrently learn image recognition and hierarchical image segmentation from unlabeled images alone. **Top:** ViT (Dosovitskiy et al., 2020) takes patch tokens as inputs and maintains the same large number of tokens through all encoder blocks. Image 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 goodness of grouping and consistency across the hierarchy. Consequently, we achieve better recognition and segmentation with higher computational efficiency.

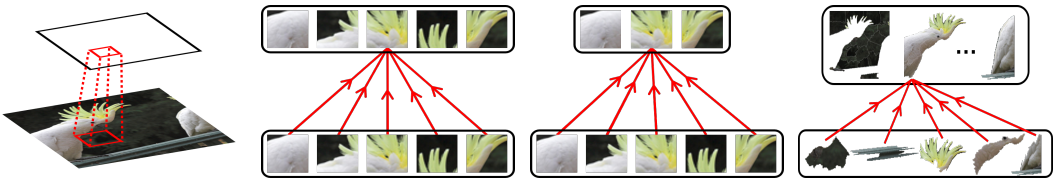
3. **Significance-based subsampling:** To increase ViT's computational efficiency, tokens are subsampled at higher levels based on their significance scores. PoWER-BERT (Goyal et al., 2020) and Token Pooling (Marin et al., 2021) define the *significance score* as the total attention given to each token from all other tokens. Downsampling then retains only 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.

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 (Dosovitskiy et al., 2020). We develop *Concurrent recognition and segmentation with Adaptive Segment Tokens* (CAST). It has three novel aspects (Fig. 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



1) CNN **2)** ViT **3)** Token Pooling **4)** Our CAST

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

Figure 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 (Krizhevsky et al., 2012; He et al., 2016) computes features on a regular grid and handles more global information by spatial downsampling. ViT (Dosovitskiy et al., 2020) takes regularly shaped patches as inputs, and updates features using attention (Vaswani et al., 2017). Token Pooling (Marin et al., 2021) 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.

unsupervised recognition by maximizing image-wise discrimination (Chen et al., 2021). 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.

2 Related Works

Vision Transformers. Vision transformers (ViT) (Dosovitskiy et al., 2020) and its followups (Touvron et al., 2021; Yuan et al., 2021; Wang et al., 2021a) adopt the transformer architecture originally for NLP proposed in (Vaswani et al., 2017). Remarkably, ViT achieves state-of-the-art performance on ImageNet1k (Deng et al., 2009) with pretraining on JFT-300M. However, its high computation costs limit its applications. Its computational complexity is derived from two factors: the latent feature dimensions and number of tokens. Many research works then tackle one of these aspects.

To reduce the latent feature computation, one direction is to restrict the attention connections by leveraging spatial relationship in the data (Parmar et al., 2018; Ramachandran et al., 2019; Beltagy et al., 2020; Child et al., 2019; Zaheer et al., 2020) or utilizing hashing, sorting, or compression (Kitaev et al., 2020; Vyas et al., 2020; Tay et al., 2020; Liu et al., 2018; Wang et al., 2020). 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 (Liu et al., 2021; Heo et al., 2021; Dong et al., 2022). The other attempts to measure the significance scores among the tokens and drop or prune tokens accordingly (Goyal et al., 2020; Rao et al., 2021; Marin et al., 2021). This camp is the most related to our work. Our work differs that tokens are not discarded but merged into coarser ones.

TCFormer (Zeng et al., 2022) and GroupViT (Xu et al., 2022) are two recently proposed hierarchical vision transformers. Both models still use fixed-shape patch tokens, resulting in noise among object boundaries. In addition, TCFormer is only demonstrated in supervised settings, and GroupViT requires text supervision. In contrast, our work gets to fine-grained pixel groupings early and develops adaptive segment tokens throughout the model, which naturally induces hierarchical image segmentations. Our model does not require any human labels.

Image Segmentation and Clustering. Image segmentation is referred to as partitioning an image by coherent regions. Classic methods have two steps: extracting local features and clustering them based on different criteria, e.g., mode-finding (Comaniciu & Meer, 2002; Banerjee et al., 2005), or graph partitioning (Felzenszwalb & Huttenlocher, 2004; Shi & Malik, 2000; Malik et al., 2001; Yu & Shi, 2003; 2004). Conventionally, a hierarchical segmentation is predicted as output for comparing against human perception (Arbelaez et al., 2010). The common approaches, to avoid ambiguities along object boundaries, typically resort to contour detection (Hwang & Liu, 2015; Xie & Tu, 2015) and eliminate contours one by one to form different levels of segmentation (OWT-UCM) (Arbelaez et al., 2010). Such approaches train on the finest level of groundtruth segmentation and hope to produce coarser levels of segmentation automatically at inference time. On the other hand, our work operates directly on segments as opposed to contour proxies.

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 (Wu et al., 2018; He et al., 2020; Chen et al., 2020; Wang et al., 2021c), adding an contrastive loss across views (Wang et al., 2021b), or by stronger augmentation and constrained cropping (Van Gansbeke et al., 2021; Selvaraju et al., 2021). **2)** A pixel-wise cluster predictor can be learned by maximizing the mutual information between cluster predictions on augmented views of the same instance at corresponding pixels (Ji et al., 2019; Ouali et al., 2020). **3)** A pixel-level feature encoder can be learned directly by maximizing discrimination between pixels based on either contour-induced segments (Hwang et al., 2019), pre-computed region hierarchies (Zhang & Maire, 2020), or hierarchical clustering transformers (Ke et al., 2022). Segmentation is thus derived from pixel feature similarities. Our work unifies the first camp and the third camp, as we train ViT with a self-supervised image recognition framework while naturally producing unsupervised hierarchical segmentation.

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. 3.

3.1 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 (Hwang et al., 2019) and HSG (Ke et al., 2022)). 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

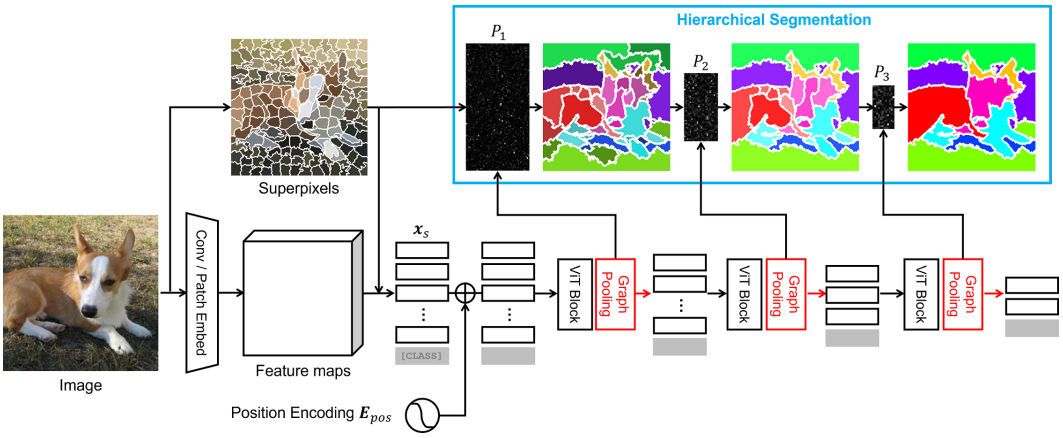


Figure 3: Our CAST jointly produces **1) a hierarchical image segmentation** and **2) corresponding multi-level features** for an input image. Building on ViT (Dosovitskiy et al., 2020), our model considers segment, not patch, tokens as inputs and outputs. We first oversegment the image into superpixels, 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 input tokens to subsequent transformer blocks. Followed by a **graph pooling** module, we group segments into coarser regions and derive coarsened tokens by aggregating 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 corresponding multi-level representations.

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 image oversegmentation methods, such as Seeds (Bergh et al., 2012), to partition an image into locally connected and color-wise coherent regions, a.k.a superpixels. Detailed in section 3.3, 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 (Dosovitskiy et al., 2020), we append $\mathbf{x}_s$ with a learnable embedding, a.k.a `[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 $\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_s] + \mathbf{E}_{pos}$.

3.2 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 thus have little knowledge of global correlation. However, segmentations are adaptive to images: every segment has different shapes. Optimal segmentation also requires group-wise correlation among image pixels (Shi & Malik, 2000). Therefore, we prefer globalized architectures, such as ViT (Dosovitskiy et al., 2020) and GNN (Kipf & Welling, 2016), over the grid-based models.

Algorithm 1: 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 $\mathbf{u} \leftarrow \mathrm{MSA}(\mathrm{LN}(\hat{\mathbf{z}})) + \hat{\mathbf{z}}$
Centroid feature $\mathbf{v} \leftarrow \{\mathbf{u}_i | i \in \mathrm{S}\}$
```
/* Assign new clustering.                 */
```
$P \leftarrow \mathrm{softmax}(\kappa \mathbf{u}\mathbf{v}^\top)$
$P_h \leftarrow \mathrm{one_hot} \circ \mathrm{arg_max}(P)$
```
/* Output new features.                   */
```
Average pooled feature $\bar{\mathbf{z}} \leftarrow P_h^\top \hat{\mathbf{z}} / P_h^\top 1$
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 2: Overall framework

Input: Initial segment token $\mathbf{x}_s$, class token $\mathbf{x}_{class}$, position encoding $\mathbf{E}_{pos}$ and grouping steps Δ
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
$\mathbf{f}_{seg} \leftarrow \mathrm{FC}(\mathrm{BN}([\mathrm{Unpool}(\mathbf{z}_l^{1:n_l})); \ldots]) \quad l \in \Delta$

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) (Vaswani et al., 2017), 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.

3.3 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. 1.

Existing methods (Xu et al., 2022; Ke et al., 2022) perform clustering with a set of learnable embeddings, resulting in fixed segmentation granularity. Yet, the optimal number of segments vary with image scales: larger (smaller) images require more (less) 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 (Qi et al., 2017) 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 average aggregating within a cluster. For detailed computation, please refer to Alg. 1.

We apply graph clustering losses (Tsitsulin et al., 2020) to cluster input tokens and optimize our graph pooling modules. At each level, we construct an affinity matrix A and the degree vector D from the pair-wise similarity among inputs. Let m and n_l be the total number of connection edges and output tokens at level l. We follow Ke et al. (2022) and apply three losses: **1) Modularity maximization**: $\mathrm{L}_l^{\mathrm{mod}} = -\frac{1}{2m}\mathrm{trace}(P_l^\top(A - \frac{1}{2m}DD^\top)P_l)$. Based on group-wise correlations among

inputs, the loss seeks an optimal partitioning of inputs, such that in-cluster (out-cluster) similarity is higher (lower) than total expectation. **2) Collapse regularization**: $\mathrm{L}_l^{\mathrm{col}} = \frac{\sqrt{n_l}}{n_{l-1}}\|1^\top P_l\|_F - 1$. The loss encourages an even (similar-sized) partitioning of inputs. **3) Coarse-region contrastive loss**: $\mathrm{L}_l^{\mathrm{crc}} = \frac{1}{n_l}\sum_i -\log\frac{\exp(\kappa\tilde{\mathbf{z}}_{l,i}^\top\tilde{\mathbf{z}}_{l,i})}{\sum_j \exp(\kappa\tilde{\mathbf{z}}_{l,i}^{l\top}\tilde{\mathbf{z}}_{l,j})}$, where $\tilde{\mathbf{z}}_{l,i} = \frac{\mathbf{z}_{l,i}}{\|\mathbf{z}_{l,i}\|}$ are the normalized coarsened tokens. The loss contrasts each coarsened region against others to increase the distinctiveness.

3.4 Training and Inference

We summarize our overall framework in Alg. 2. We conduct segment grouping at certain levels in the model. We train our CAST using an image-wise contrastive learning framework–Moco-v3 (Chen et al., 2021). The objective is to contrast each image against others in the latent feature space. Let $\mathrm{L}^{\mathrm{moco}}$ denote the corresponding loss. Our overall loss is: $\mathrm{L} = \lambda_m \mathrm{L}^{\mathrm{moco}} + \lambda_g \sum_{l\in\Delta}(\mathrm{L}_l^{\mathrm{mod}} + \mathrm{L}_l^{\mathrm{col}}) + \lambda_c \mathrm{L}^{\mathrm{crc}}$. We only impose the coarse-region contrastive loss in the final graph pooling module.

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 (Hariharan et al., 2015) and unpool higher-level features based on the grouping index w.r.t superpixels.

4 Experiments

Datasets. **1) ImageNet** (Deng et al., 2009) 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 mostly locate at the image center. The training and validation set includes 1.28M and 50K images, respectively. We follow Tian et al. (2020) to subsample 100 object categories to create IN-100. The subset consists of 127K and 5K images for training and testing. **2) Pascal VOC 2012** (Everingham et al., 2010) is an object-centric semantic segmentation dataset that contains 20 object categories and a background class. We use the augmented training set (Hariharan et al., 2011) with $10,582$ images and validation set with $1,449$ images. **3) MSCOCO** (Lin et al., 2014) is a generic scene parsing dataset. The scene contexts are more complex and include more objects in each image (7.3 vs. 2.3) than VOC. Following Van Gansbeke et al. (2021), we train on $118,287$ images of *train2017* split and test on the VOC validation set.

Model architecture. For most experiments, we base our architecture on ViT-S (Dosovitskiy et al., 2020), which has 384 channel dimensions of all encoder blocks. We follow Xiao et al. (2021) 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 (Marin et al., 2021). We thus re-implement the method by combining PoWER-BERT (Goyal et al., 2020) with K-Medoids clustering algorithm. We follow the same settings as our CAST.

Image resolution and token numbers. For training on all datasets and testing on ImageNet, we set $\mathrm{crop_size}$ 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.

Model training. Based on MoCo, we train all models from scratch without any human-labeled supervision. Mostly follow MoCo, we set $\mathrm{batch_size}$ to 256, $\mathrm{learning_rate}$ to $1.5e^{-4}$, $\mathrm{weight_decay}$ to 0.1, and $\mathrm{momentum}$ to 0.9. We use AdamW (Kingma & Ba, 2014) optimizer. For hyper-parameters of MoCo framework, we set $\mathrm{temperature}$ to 0.2, output dimension to 256, $\mathrm{momentum_coefficients}$ 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 $\mathrm{warmup_epochs}$ to 20, 10 and 40, respectively. Cosine decay schedule is applied to adjust the learning rate. We set λ_m, λ_g, λ_c to 1.0, 0.4, and 0.1.

The gradients calculated from the modularity maximization and collapse regularization loss are used to update only our graph pooling modules. We do not propagate the gradients to the rest of the model.

Model testing. **1) 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 the classifier. We train the linear classifier with ground-truth labels and report Top-1 accuracy. **2) On VOC**, we follow Van Gansbeke et al. (2021) 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 the training set and testing on the validation set. **3) On figure-ground segmentation**, we binarize multi-head attention maps: Following DINO (Caron et al., 2021) 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. See Appendix for more details.

backbone	method	# tokens	GFLOPS	IN-100	IN-1k
ViT-S	Vanilla	$[196] \times 4$	9.27	78.1	**67.9**
	K-Medoids	$196, 64, 32, 16$	**5.34**	75.8	62.8
	Our CAST	$196, 64, 32, 16$	5.88	**78.4**	66.1
ViT-B	Vanilla	$[196] \times 4$	38.4	81.7	-
	K-Medoids	$196, 64, 32, 16$	**20.8**	81.6	-
	Our CAST	$196, 64, 32, 16$	22.7	**82.0**	-

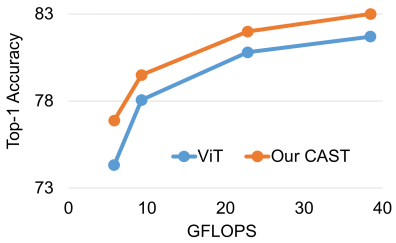


Table 1: Our model achieves a better trade-off between model efficiency and task performance for unsupervised image classification on ImageNet val set. We report the top-1 accuracy of the linear classifier. **Left:** Image classification on IN-100 and IN-1k. Compared to vanilla ViT, our CAST reduces 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.

Result 1: Better accuracy on ImageNet 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 1). On IN-100, using ViT-S and ViT-B backbone, our CAST achieves comparable performance as vanilla ViT, yet our model is 63.4% and 59.1% 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 (Marin et al., 2021) or model distillation (Rao et al., 2021) 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 1). Our CAST outperforms vanilla ViT using the same computational budget.

Trained on MSCOCO, to be tuned on VOC			before		after	
method	# tokens	GFLOPS	mIoU	F-score	mIoU	F-score
Vanilla ViT	$[1024] \times 4$	65.4	30.9	16.1	65.8	40.7
K-Medoids	$1024, 320, 160, 80$	38.4	27.6	17.2	66.7	47.5
Our CAST	$1024, 320, 160, 80$	42.7	**32.7**	**21.3**	**66.8**	**48.2**

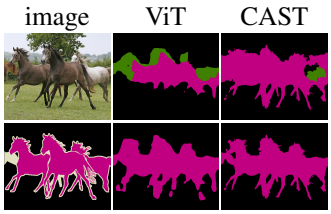


Table 2: Our predicted segmentations are much more precise and better aligned with ground-truth (row 2 column 1 image) semantic boundaries on VOC val set, benchmarked with the regional mIoU and boundary F-score metric. Segmentations are predicted based on segment-wise nearest neighbor retrievals (before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

Result 2: Better semantic segmentation on VOC. We evaluate unsupervised semantic segmentation on VOC in Table 2. 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 ($+1.8\%$ and $+1.0\%$ mIoU) and respect object boundaries better ($+3.9\%$ and $+5.1\%$ F-measure) than baselines, yet model efficiency is greatly improved.

Result 3: Better figure-ground segmentation on VOC. 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 3, our CAST outperforms DINO and ViT by $+2.1\%$. Our foreground masks can cover dominant foreground objects and better align with object boundaries.

method	IoU
Our CAST	**48.0**
Vanilla ViT	45.8
DINO	45.9

Table 3: 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. **Left)** Jaccard similarity between ground-truth and predicted foreground masks. **Right)** Visual comparisong between our CAST (top row) and ViT (bottom row). Our predicted masks are more coherent within foreground regions and better align with object boundaries.

Result 4: Better hierarchical segmentation on IN-1K. Fig. 4 shows that our hierarchical segmentations respect semantics at different levels of granularity better than ViT and K-Medoids baselines. Our model directly produces hierarchical segmentations without needing additional clustering.

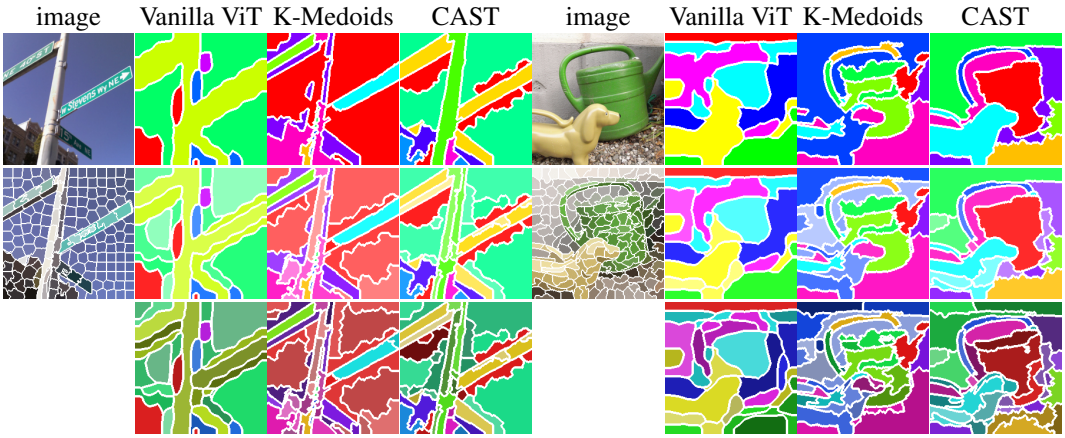


Figure 4: Our CAST generates higher-quality hierarchical segmentation. From left to right, we show the input image and hierarchical segmentations generated by vanilla ViT, K-Medoids clustering, and our CAST. We also show the superpixels (white contours) generated from input images. From top to bottom, 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.

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.

REFERENCES

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

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(Sep): 1345–1382, 2005.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *European conference on computer vision*, pp. 13–26. Springer, 2012.

Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

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*, pp. 9640–9649, 2021.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

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

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12124–12134, 2022.

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

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pp. 3690–3699. PMLR, 2020.

Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pp. 991–998. IEEE, 2011.

Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 447–456, 2015.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11936–11945, 2021.

Jyh-Jing Hwang and Tyng-Luh Liu. Pixel-wise deep learning for contour detection. *arXiv preprint arXiv:1504.01989*, 2015.

Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *ICCV*, 2019.

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*, pp. 9865–9874, 2019.

Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2571–2581, 2022.

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

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *IJCV*, 2001.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860*, 2021.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pp. 416–423, July 2001.

Yassine Ouali, Celine Hudelot, and Myriam Tami. Autoregressive unsupervised image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pp. 4055–4064. PMLR, 2018.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34, 2021.

Ramprasaath R Selvaraju, Karan Desai, Justin Johnson, and Nikhil Naik. Casting your model: Learning to localize improves self-supervised representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11058–11067, 2021.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning*, pp. 9438–9447. PMLR, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *arXiv preprint arXiv:2006.16904*, 2020.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Revisiting contrastive methods for unsupervised learning of visual representations. *arXiv preprint arXiv:2106.05967*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. *Advances in Neural Information Processing Systems*, 33:21665–21674, 2020.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, 2021a.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3024–3033, 2021b.

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*, pp. 12586–12595, 2021c.

Andrew P Witkin and Jay M Tenenbaum. On the role of structure in vision. In *Human and machine vision*, pp. 481–543. Elsevier, 1983.

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*, 2018.

Tete Xiao, Piotr Dollar, Mannat Singh, Eric Mintun, Trevor Darrell, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34, 2021.

Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015.

Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18134–18144, 2022.

Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *ICCV*, 2003.

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

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 558–567, 2021.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. 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*, pp. 11101–11111, 2022.

Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. *Advances in Neural Information Processing Systems*, 33, 2020.

A APPENDIX

We develop the first vision transformer that *concurrently* achieves image-wise recognition and hierarchical image segmentation without additional processing. Our CAST outperforms existing token coarsening methods for image classification, semantic segmentation, and attention-induced figure-ground segmentation tasks. In this supplementary, we include more details as followings:

- We present more visual results of hierarchical segmentation on ImageNet in A.1.
- We present more visual results of attention-induced figure-ground segmentations on VOC in A.2.
- We present visual results of attention maps from our CAST on ImageNet in A.3.
- We present the ablation study regarding the proposed regularizations and graph pooling module in A.4.
- We present the ablation study on inference latency of our CAST in A.5.
- We present the ablation study on effectiveness of superpixels in A.6
- We present quantitative results of hierarchical segmentation on DensePose in A.7
- We illustrate more details of inference in A.8.

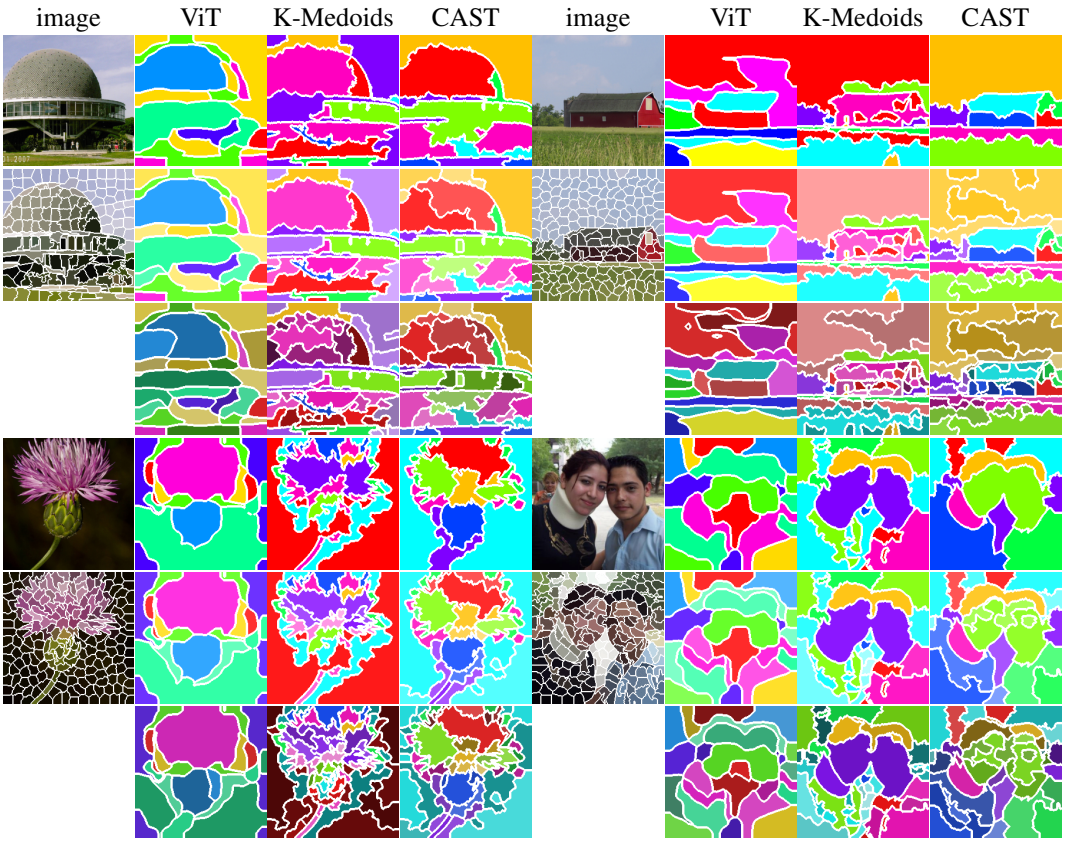


Figure 5: Our CAST generates higher-quality hierarchical image segmentation. From left to right, we show the input image and hierarchical segmentations generated by ViT, K-Medoids clustering, and our CAST. We also show corresponding superpixels (white contours) generated from input images. From top to bottom, we hierarchically segment the input image into 8, 16, and 32 regions. ViT requires additional pixel-wise clustering (e.g. K-Means) on fixed feature representations, yet the results are over-smoothed. Our CAST naturally generates hierarchical segmentations without any post-processing. Our segmentations align with image boundaries and capture semantics more precisely.

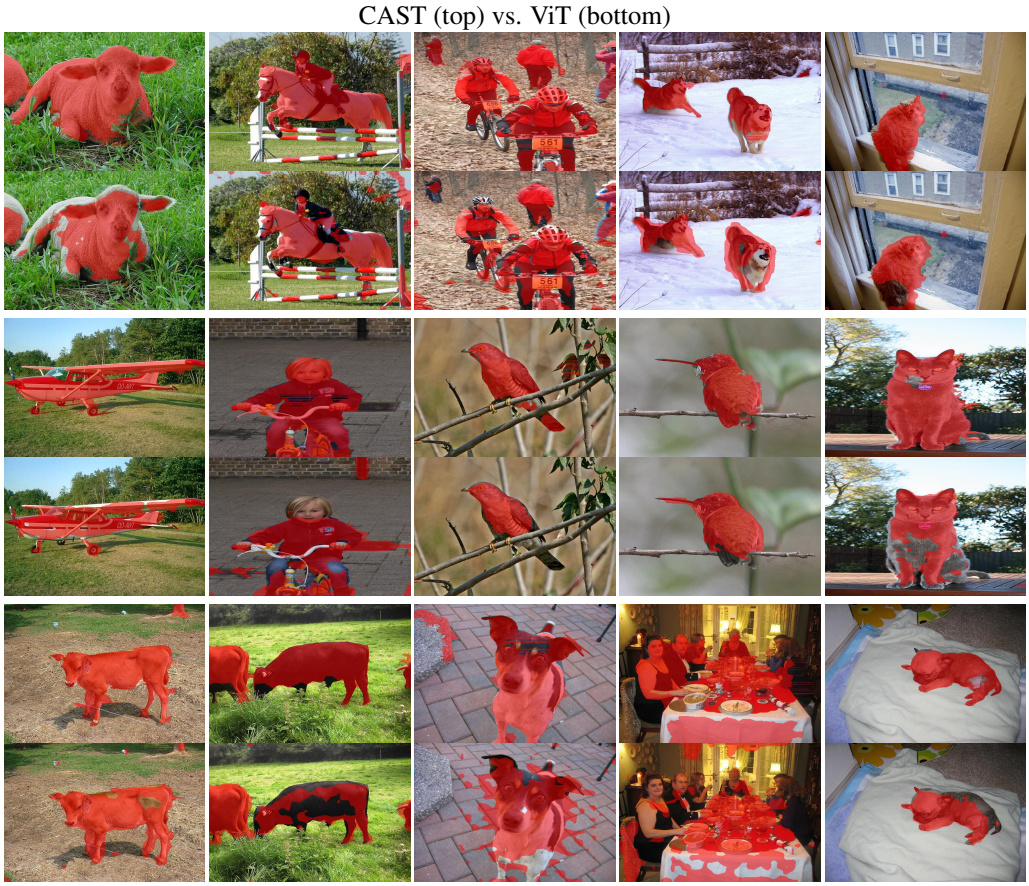


Figure 6: Our CAST (top row) attends to foreground semantics more precisely than vanilla ViT (bottom row) and DINO (Caron et al., 2021) on VOC. We adopt the same procedure as DINO to generate foreground segmentation masks from latent multi-head attention maps. All models are trained on IN-1K dataset from scratch. Our CAST and ViT are trained based on MoCo-v3 (Chen et al., 2021).

A.1 Visual Results on Hierarchical Segmentation.

We present more visualization results of hierarchical segmentations induced by vanilla ViT, K-Medoids, and our CAST (see Fig. 5). We hierarchically segment the input image into 32, 16, and 8 regions. Notably, Vit baseline requires additional pixel-wise K-Means clustering on fixed feature representations, yet the results are over-smoothed. Our CAST naturally generates hierarchical segmentations without any post-processing. Our multi-scale segmentations align with image boundaries and capture semantics more precisely.

A.2 Visual Results on Attention-induced Figure-ground Segmentation.

We present more visual results of figure-ground segmentations generated from multi-head attention maps on VOC. Our CAST attends to foreground semantics more precisely than vanilla ViT, and the segmentations preserve object boundaries more accurately. See Fig. 6.

A.3 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

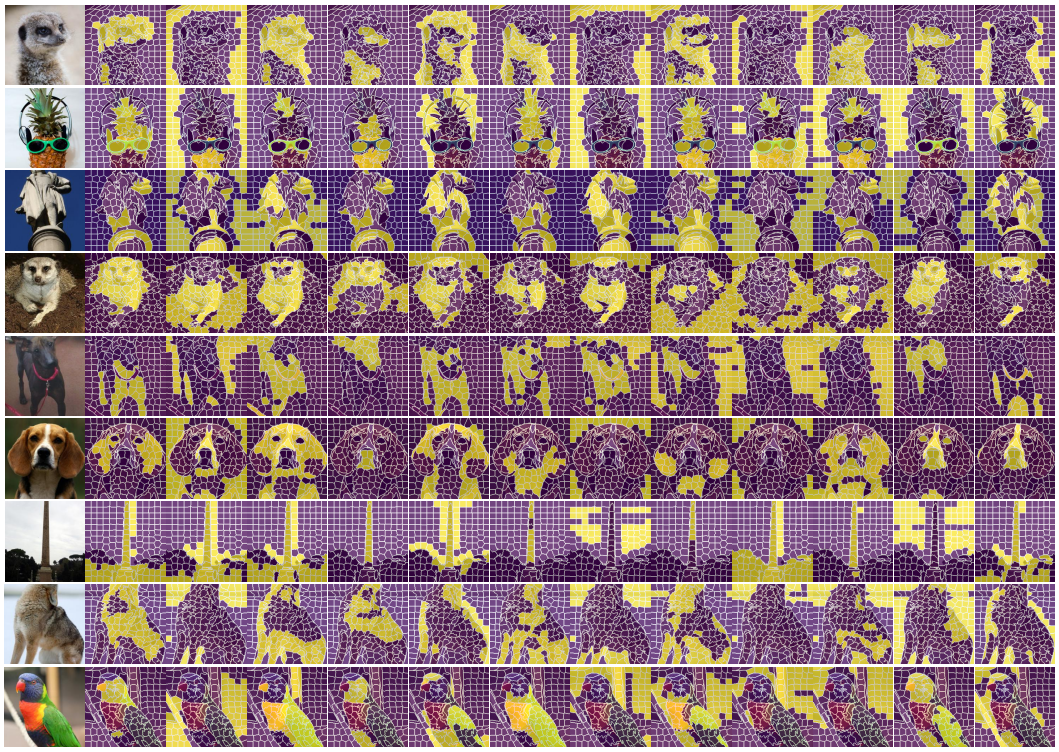


Figure 7: Our multi-head attention maps reveal parts-of-the-whole information of the image on IN-100. **From left to right:** input images and corresponding 12 heads of attention maps of the `[CLASS]` token to all the other segments. We follow DINO (Caron et al., 2021) to binarize attention maps. We show that the 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 attention maps better aligned with object boundaries.

the figure-ground segmentation, which are the ones in the 9th 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 (Caron et al., 2021) to display the binarized attention maps. The threshold is adjusted to keep 60% of the mass. See Caron et al. (2021) for more details.

As shown in Fig. 7, 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	λ_m	λ_g	λ_c	Acc.
Patch	-	1.0	0.0	0.0	78.1
Segment	-	1.0	0.0	0.0	78.1
Segment	✓	1.0	0.0	0.0	78.0
Segment	✓	1.0	0.4	0.0	78.3
Segment	✓	1.0	0.4	0.1	78.4

Pooling	Acc.
Our Graph Pooling	78.4
K-Medoids: PoWER-BERT	75.8
K-Means: PoWER-BERT	73.9
K-Medoids: Random Sampling	72.3

Table 4: Our proposed graph pooling module and loss regularizations improve image classification on IN-100 val set. **Left:** Improved performance by adding each loss regularization. **Right:** Our graph pooling module outperforms K-Medoids and K-Means clustering algorithm by a large margin. Cluster centroids are initiated by either PoWER-BERT or random sampling.

A.4 Ablation Study on Proposed Regularizations and Graph Pooling Module.

Summarized in Table 4, we demonstrate the efficacy of the proposed loss regularizations and graph pooling module. We report top-1 accuracy results on IN-100 dataset. We show that we improve our model's performance by adding each proposed loss. 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 5: FPS in our graph pool module requires additional computation. We report the inference time (ms) of each module with 384 channel dimensions and 256 batch sizes on IN-100. Optimizing the token sampling technique to increase model efficiency is our future work.

A.5 Ablation Study on 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 5, 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.

A.6 Ablation Study on Superpixel

We next verify the superior efficacy of superpixel tokens on dense pixel applications. As shown in Table 6, 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.

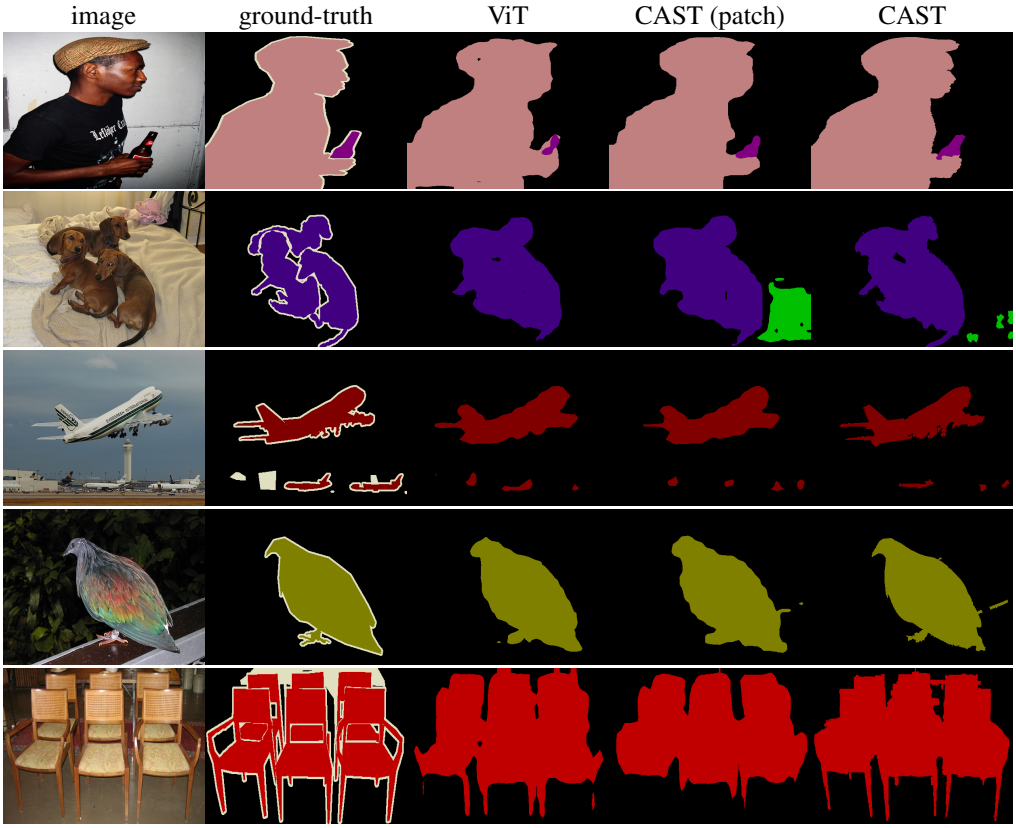
A.7 Quantitative Results on Hierarchical Segmentation

To demonstrate the efficacy of our hierarchical segmentation results, we compare CAST with K-Medoids on the DensePose dataset (Alp Güler et al., 2018). 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 (Arbelaez et al., 2010) with ground truths by F1-score. As shown in Fig. 8, our CAST outperforms K-Medoids and is better at capturing semantics at every level of granularity.

A.8 Inference and Testing.

For image classification, we apply the linear probing procedure for evaluation. For semantic segmentation, we use the segment retrieval and transfer learning procedure to evaluate model performance.

Image classification: linear probing. We follow MoCo-v3 (Chen et al., 2021) 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 Chen et al. (2021), 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



				Classification	Segmentation	
Method	GFLOPS	dim	Token	Acc.	mIoU	f-score
Vanilla ViT	65.4	384	Patch	78.1	65.8	40.7
			Segment	78.1	66.5	46.7
Our CAST	42.7	384	Patch	77.9	66.2	41.2
			Segment	**78.4**	**66.8**	**48.2**

Table 6: Using superpixel tokens improves the 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 classification (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 dimensions. For vanilla ViT and our CAST, we observe significant performance gain for semantic segmentation, though the performance gap for classification is negligible. In particular, superpixel-based methods preserve thin structures (boundaries) of objects much better than their patch-based counterparts.

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 Hwang et al. (2019); Van Gansbeke et al. (2021); Ke et al. (2022) 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 the 20 retrieved segments.

For ViT baselines, 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.

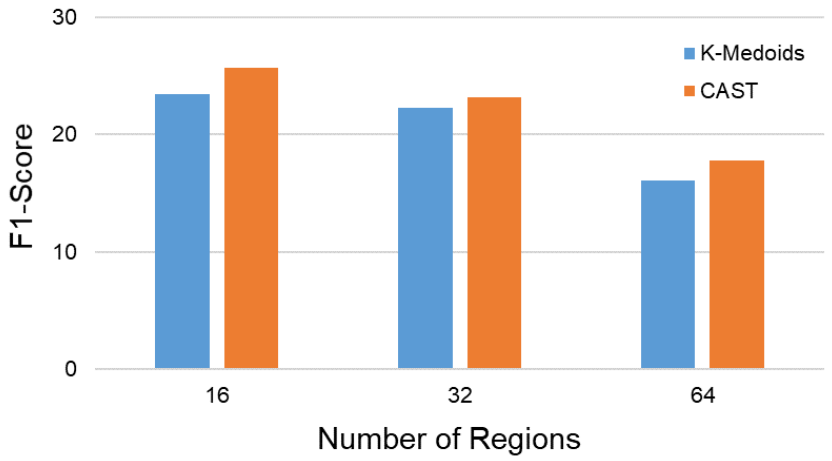


Figure 8: Our CAST delivers better hierarchical segmentation than K-Medoids clustering algorithm. On DensePose (Alp Güler et al., 2018), we parse body-part-level labels into head, torso, upper and lower limb regions. Using CAST or K-Medoids, we partition an image into 64, 32 and 16 regions, and evaluate the region-wise converings with ground-truths by F1-score. Our CAST captures semantics better at every level of granularity.

Our CAST does not require additional upsampling and K-Means clustering. For segmentation, our model follows Hypercolumn design (Hariharan et al., 2015) to unpool and fuse multi-level segment tokens. Our model generates the same number of output tokens as the superpixels. We gather pixel features from output tokens based on the superpixel index. Without the need for 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 Van Gansbeke et al. (2021) 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, $\mathrm{learning_rate}$ to 0.003, $\mathrm{weight_decay}$ to 0.0001, $\mathrm{batch_size}$ to 16, $\mathrm{crop_size}$ to 512. Following Chen et al. (2016), we adopt poly learning rate policy by multiplying base learning rate by $1 - \frac{iter}{max_iter}^{0.9}$. We adopt the SGD optimizer. We use only a single-scale image for inference.