Framework-agnostic Semantically-aware Global Reasoning for Segmentation

Mir Rayat Imtiaz Hossain[1,2] Leonid Sigal[1,2,3] James J. Little[1]
[1]University of British Columbia [2]Vector Institute for AI [3]Canada CIFAR AI Chair
{rayat137, lsigal, little}@cs.ubc.ca

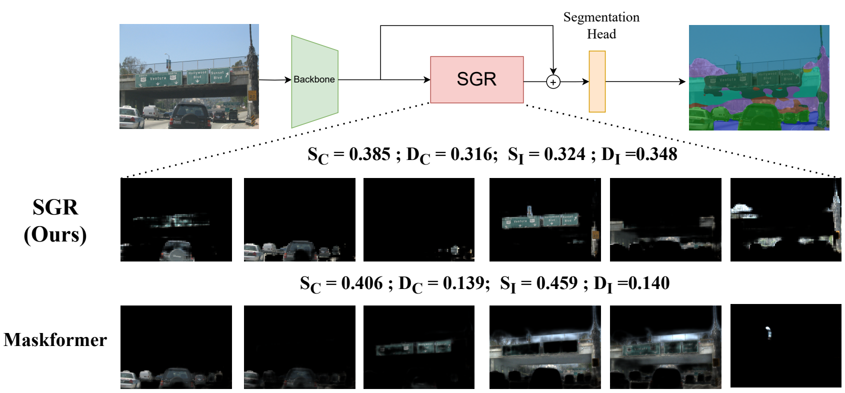

Figure 1. SGR learns to group pixels into latent tokens based on semantic similarity (top); the tokens are refined using a Transformer and back-projected to enhance the original features for segmentation. Notably, the resulting tokens are more semantic than prior work, as measured by the proposed class-semantics (S_C) and instance-semantics (S_I) metrics (↓ is better), and are more diverse (↑ is better), as measured by proposed diversity metric at class(D_C) and instance-level(D_I) (see Maskformer [19] in the bottom).

Abstract

Recent advances in pixel-level tasks (e.g. segmentation) illustrate the benefit of of long-range interactions between aggregated region-based representations that can enhance local features. However, such aggregated representations, often in the form of attention, fail to model the underlying semantics of the scene (e.g. individual objects and, by extension, their interactions). In this work, we address the issue by proposing a component that learns to project image features into latent representations and reason between them using a transformer encoder to generate contextualized and scene-consistent representations which are fused with original image features. Our design encourages the latent regions to represent semantic concepts by ensuring that the activated regions are spatially disjoint and the union of such regions corresponds to a connected object segment. The proposed semantic global reasoning (SGR) component is end-to-end trainable and can be easily added to a wide variety of backbones (CNN or transformer-based) and segmentation heads (per-pixel or mask classification) to consistently improve the segmentation results on different datasets. In addition, our latent tokens are semantically interpretable and diverse and provide a rich set of features that can be transferred to downstream tasks like object detection and segmentation, with improved performance. Furthermore, we also proposed metrics to quantify the semantics of latent tokens at both class & instance level.

1. Introduction

Pixel-level tasks, such as semantic [3, 12, 18, 19, 32, 39, 56, 59, 64, 70], instance [19, 19, 27], and panoptic [18, 19, 35, 36] segmentation, are fundamental to many computer vision problems (autonomous driving is a prime example). Recent studies [17, 19, 25, 61] have demonstrated that even though predictions of these tasks are essentially local, incorporating global context can significantly improve performance.

Although early approaches focused on local multi-scale context where attention is computed over progressively larger patches or adding global context by pooling (*e.g.* Parsenet [44]), recent approaches have gravitated towards attentional pooling of information into latent *tokens*[1] (*e.g.* using double-attention like in A^2-Net [16] or ACFNet [64]) and using them to enhance and contextualize pixel representations. Other approaches like PerceiverNet [31] or DynaPerceiver [26] aim at learning a dictionary of latent codes of concepts and perform a cross-attention between the latent codes and image features. The main difference between these approaches lies in how the tokens are formed, whether there are interactions between them (*e.g.*, using graph propagation [17, 40]) and how they are aggregated back to enhance the pixel representations. Although these approaches are motivated by the overarching idea that the tokens would aggregate information over individual objects, enhancing

[1]We define *latent tokens* as feature representations of, not necessarily spatially contiguous, image regions.

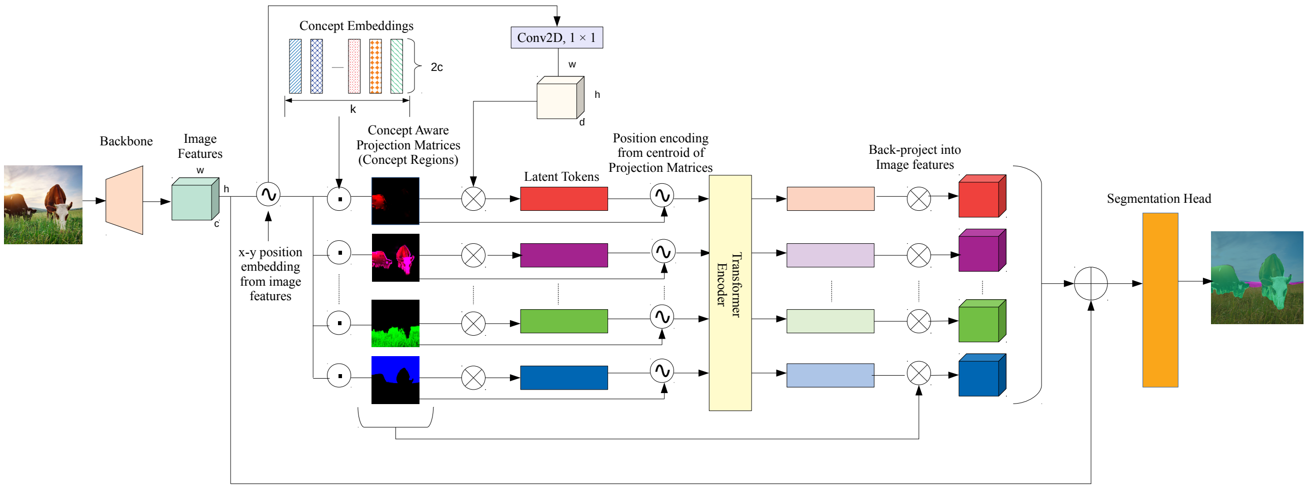

Figure 2. **Overview of our framework**. A dot-product is performed between K learned concept embeddings (randomly initialized) and position-aware image features with added x-y positional embedding to generate soft projection matrices (*concept regions*). The projection matrices aggregate features in an object-centric manner to create latent tokens. A two-layer transformer encoder is used to reason between the tokens which are re-projected to original feature space for segmentation.

pixels with object-centric globally-contextualized representations (*e.g*., through interaction between the region representations), in practice, visualizations demonstrate that the resulting latent tokens fail to capture the semantics of the scene. To address this, recent approaches have proposed to supervise the tokens directly *e.g*. OCR [62], Maskformer [19] and Mask2Former [18] use ground truth semantic mask annotations. Class-attention plays a similar role by learning class token embeddings [57]. While they result in semantically meaningful tokens and improve performance in semantic segmentation, they are limited in multiple ways. The number of tokens in these approaches [18, 19, 61] is tied to the number of classes with each token modelling the union of all instances of a particular object in the scene. In other word, they are class-, not object-centric which can be suboptimal since instances of an object may be different in appearance or shape or be located in different disjoint regions. Indiscernibly aggregating them may result in loss of detail. We posit that a more object-centric association (*e.g*. where regions may be closely associated with individual instances or *concepts* that compose those instances) and an interaction between the tokens would provide a more granular and interpretable mechanism for attentional context.

The core challenge for such object-centric aggregation lies in not using any instance supervision and solely relying on semantic supervision, like prior work. For the rest of the paper, we refer to the projection matrices (see Figure 2) which aggregate features into latent tokens as *concept regions* because they aim to group features that belong to the same concept. To this end, we make following observations: **(1)** disconnected semantic segments are likely to belong to different instances – giving us a lower bound on the number of *concept regions* per image during training; **(2)** union of *concept regions* for a particular class should correspond to the whole semantic segmentation for the class; and **(3)** each *concept region* must be spatially disjoint. These constraints allow us to formulate a rich set of objectives that encourage the object-centric aggregation of features into latent tokens and refine them to add global context between the concepts, thereby enhancing the features. We show that our proposed component can be plugged into any semantic segmentation framework, regardless of the type of segmentation losses used and leads to improved performance. Additionally, the more object-centric nature of the aggregation ensures that our model learns richer features that when transferred to object detection and segmentation tasks improve performance.

Contributions. Our contributions are as follows: **(i)** We propose a framework for semantically-enhanced global reasoning (SGR) that enhances local feature representations by learning to aggregate semantically similar local features into latent tokens. These tokens are then globally refined, using a Transformer Encoder [52], and re-projected into original representation. **(ii)** We propose a rich set of losses that encourage *concept regions* to be semantically meaningful. Specifically, we ensure that the *concept regions* are disjoint and the unions of them map to connected components of ground truth segments. **(iii)** We define new metrics that measure class- and instance-level semantics of the latent tokens by considering the entropy over ground truth labels that form each *concept region*. **(iv)** Our component is agnostic to the backbone (CNN or transformer-based) or segmentation head and can be easily plugged into any segmentation framework. **(v)** We experimentally demonstrate that adding our model to different segmentation frameworks consistently increases performance across three different benchmark datasets: Coco-Stuffs-10K, ADE-20K and Cityscapes regardless of per-pixel classification or mask-classification approach. **(vi)** Lastly, the more semantically meaningful and diverse latent representations lead to a richer feature space that improves performance when transferred to downstream tasks of object detection and segmentation.

2. Related Work

Semantic Segmentation. Semantic segmentation has been approached by various methods in the past, including using graph-cuts [4, 5, 48] over specified seed points to group similar pixels and by generating mask proposals [1, 2, 10, 51] and classifying them [9, 21]. Advances in deep learning have led to formulations of segmentation problem as a per-pixel classification [12, 13, 46]. Recent studies [13–15,17,32,40,56,62] have shown that incorporating global context improves performance of semantic segmentation. More recently, approaches like Maskformer [19] and Mask2former [19] revived earlier mask-classification ideas, by generating mask queries and classifying them.

Capturing Global Context. Earlier approaches to capture the global context included increasing receptive fields [13–15] using atrous convolution [22, 29, 60] or by aggregating features from multiple scales [15, 59, 67].

Global Reasoning. Graph-based approaches and self-attention have been widely used for global reasoning between different concepts. CRFs [24, 37] were previously used for modeling pairwise interaction between the pixels in an image, particularly to refine predictions [3, 11, 12]. Non-local Nets [53] and variants built fully connected graphs over all the pixels or a set of sampled pixels [65] and used self-attention over the graph nodes. Recently, some approaches [17, 40, 41, 54, 66] densely project the image features into latent nodes and then reason between them using graph convolutions [17, 34, 40] or transformers [52, 54]. However, visualization of their latent spaces suggests failure to capture object semantics at either class or instance level. To address this, [18, 19, 61] directly supervise latent representations. Other approaches [25,30,62] used different forms of self-attention [52] to obtain the pair-wise relationship between each pixel and aggregate the information to capture the global context. Vision Transformers [23] and variants [8, 32, 45, 49, 56, 68] have achieved state-of-the-art results on various computer vision tasks, including semantic segmentation, by dividing images into patches, similar to words in language models, and applying self-attention [52] to reason between them. Subsequent approaches applied shifted windows [45] or clustering [63] to group the patches. Recently Dino [8] observed that the attention maps of class tokens tend to attend to specific regions. MCT [58] extended [8] into multiple-class tokens to make them class-aware.

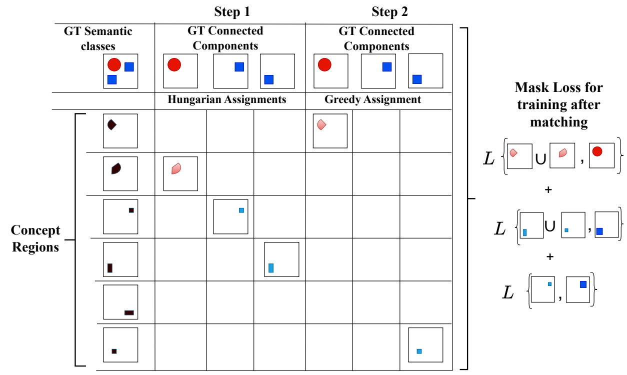

Figure 3. **Visualization of our greedy matching strategy.** Binary mask losses are used to compute a cost matrix, based on which, Hungarian matching is applied to perform 1-to-1 matching between the *concept regions* and the ground truth components. Next, we greedily select L-top matches from the remaining ones. Once matched, in training, we compute losses between the union of predicted masks assigned to the same connected component.

3. Method

The motivation for our approach is simple: to enhance the local features with global contextualized scene information. To this end, we introduce a semantically enhanced global reasoning (SGR) component shown in Fig. 2. The input to our component is a feature map from a backbone (CNN or transformer-based) with dimensions $W \times H \times C$, where W and H are spatial dimensions and C is the channel dimension. The output is a globally contextualized and enhanced feature map of the same size.

SGR consists of four main steps. First, we compute K soft masks, which we refer to as *concept regions*, by calculating the similarity between input features and a set of learned concept embeddings. The concept regions are weakly supervised by connected components of ground truth semantic labels. Next, the K *concept regions* are used to project features into K latent tokens, similar to context vector aggregation in soft attention. A positional embedding is added to the tokens based on the centroids of the corresponding *concept region*. It helps in concept disambiguation (*e.g.*, a low-textured blue token in the lower- and upper-part of the image can be disambiguated, permitting distinctions between `sky` and `water`). Third, a Transformer Encoder [52] is used to globally reason between the latent tokens. Finally, we re-project the refined tokens to the input feature map using the same *concept regions*.

The key to ensuring the "semanticness" of our *concept regions*, and hence tokens, is in the first step where the K concept embeddings are learned across the dataset. The concept embeddings are randomly initialized and in practice implemented by 1×1 convolution. Since we want our tokens to ideally represent object instances or even parts of the instances, the number of concepts we learn must be substantially larger than the number of object classes (unlike [17, 19]). Since we do not use instance-level annotations, we use connected components over semantic segmentation annotations to provide a *lower* bound to the number of concepts that should be active in a given training image. We assume that a subset of L (of K) concepts can be active in each image and guide the learning of concept embeddings by matching the *concept regions*, that drive the latent tokens, to the class-specific connected components using a greedy matching strategy (similar to MDETR [33]). Since connected components are the *lower* bound, the matching from *concept regions* to connected components is many-to-one (*i.e.*, multiple *concept regions* can be matched to a connected component). The supervision is such that the union of the *concept regions* that are matched to a connected component corresponds to the entire component. To ensure spatial diversity of the *concept regions* matched to a particular component, we minimize the pairwise cosine similarity between them.

3.1. Projection to Latent Semantic Tokens

The process of mapping image features into latent tokens is illustrated in Figure 2. Given an input feature map from a backbone, $\mathbf{X} \in \mathbb{R}^{W \times H \times C}$, where W, H and C are width, height and number of channels respectively, we add positional embeddings $PE_{pos}(X)$ and $PE_{pos}(Y)$. This is done by computing sines and cosines of different frequencies for x- and y- axes of each feature cell where $PE_{pos}(x) \in \mathbb{R}^C$. The result is a positionally aware local feature tensor:

$$\mathbf{X}' = \begin{bmatrix} \mathbf{X} + PE_{pos}(X) \\ \mathbf{X} + PE_{pos}(Y) \end{bmatrix}, \tag{1}$$

where $\mathbf{X}' \in \mathbb{R}^{W \times H \times 2C}$. This allows the module to distinguish between features which are visually similar but are at different locations.

The *concept regions* are obtained by computed dot product between each feature cell of $\mathbf{X}'$ with learned concept embeddings $\mathbf{b}_k \in \mathbb{R}^{2C}$; $k \in [1, K]$ where K is the total number of latent tokens. The concept embeddings are randomly initialized and are implemented as 2D convolution with K kernels of size 1×1. The resulting *concept regions* are K soft masks of resolution $W \times H$, collectively forming $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_K] \in \mathbb{R}^{W \times H \times K}$:

$$\mathbf{P} = \texttt{sigmoid}(\texttt{Conv2D}(\mathbf{X}'; \{\mathbf{b}_k\}_{k=0}^{K})). \tag{2}$$

Another 1×1 convolution layer is used to reduce the dimensionality of $\mathbf{X}'$ to $\mathbb{R}^{H \times W \times D}$. The K latent tokens are formed by matrix multiplication between the dimensionally reduced positional features and obtained *concept regions*:

$$\mathbf{T} = \mathbf{P}^T \odot \texttt{Conv2D}(\mathbf{X}'; \mathbf{W}_d)^T. \tag{3}$$

3.2. Global Reasoning Between the Tokens

Once we obtain latent tokens, we add positional encoding for the location of the tokens themselves into their respective representations. We characterize the position of the token by a weighted centroid computed by its concept region. Mainly, $\mathbf{T}' = \mathbf{T} + PE_{pos}(\mathbf{C})$, where

$$\mathbf{C}_k = \frac{1}{\sum_i \sum_j \mathbf{P}_{i,j,k}} \sum_{w=1}^{W} \sum_{h=1}^{H} [w\mathbf{P}_{w,h,k}; h\mathbf{P}_{w,h,k}]\,. \tag{4}$$

The tokens with positional encoding, $\mathbf{T}'$, are passed to a simple *two-layer* transformer encoder [52] which applies self-attention over the tokens thereby performing global reasoning between them. The weights of this transformer encoder block are randomly initialized and not pre-trained. Overall, this reasoning is analogous to identifying the relationships between different semantic concepts at different locations because each of the latent tokens correspond to a connected component of a concept.

The output from the transformer is then re-projected onto the feature space using the same concept regions that we used to project the original features. The re-projected features are added with the input features and passed to the segmentation head.

3.3. Concept Region Supervision

We aim to make each token semantic, such that it only aggregates features from the component of the semantic class (ideally an instance, but possibly a part of the instance of an object) it represents. For example, we would like to have concepts that represent each *car* in a scene, but would also be happy with concepts that separately represent each car's *wheels*, *body* and so forth. We observe that disconnected components of the semantic segments for a given object class are likely to belong to different instances (see supplementary) and hence they should correspond to different concepts as per the construction above. To obtain connected component ground truth for supervision, we apply connected component analysis over ground truth segmentation masks.[2]. During training, we first match the concept region of each token to these connected components based on a cost matrix. Once matched, we supervise the concept regions using the ground truth connected components to ensure that tokens are spatially contiguous and semantic.

Specifically, given K concept regions, we assume that only up to L of them are active in any given image[3] and must be matched to C ground truth connected components.

[2]We apply simple morphological operations and ignore components that are smaller than 5% of the maximum area of the connected components in the given training image.

[3]Note that K concepts are shared for the dataset and only a subset of those are likely to be present in any one image. We let L be the maximum number of concepts that are present in an image.

We assume $K > L > C$ (for example in most experiments we let $K = 512$ and $L = 64$). We first form a $K \times C$ cost matrix and match using a two-stage procedure. First, we use the Hungarian algorithm [7, 19, 38] to perform a bipartite matching between ground truth connected components and concept regions. This ensures that each connected component is associated with at least one concept region. Second, we match the remaining $L - C$ concept regions greedily to connected components by considering the remaining $(K - C) \times C$ portion of the cost matrix. The procedure is illustrated in detail in Figure 3.

Each (i, j)-th entry in the cost matrix measures similarity between the i-th *concept region* $\mathbf{P}_i$ and the binary ground truth mask $\mathbf{M}_j$ for j-th connected component. To measure similarity we use a combination of dice loss [47] and focal loss [42],

$$Cost_{i,j}(\mathbf{M}_j) = \mathcal{L}_{focal}(\mathbf{P}_i, \mathbf{M}_j) + \rho\mathcal{L}_{dice}(\mathbf{P}_i, \mathbf{M}_j). \quad (5)$$

The hyperparameter ρ controls the relative weight of the two terms in the cost computation.

Once matched, we can supervise the concept regions using the same losses used to match them. However, many-to-one matching between concept regions and ground truth connected components may lead to tokens that are duplicates to others. To avoid this, we add regularization which ensures that concept regions compete for support. This is achieved by cosine similarity loss that is applied to pairs of concept regions that are matched to the *same* connected component: $\mathcal{L}_{cos}(\mathbf{P}_{i\to j}, \mathbf{P}_{k\to j})$. The final loss for supervision of latent concepts can be written as follows:

$$\mathcal{L}_{concept}(\mathbf{P}, \mathbf{M}) = \sum_{j=1}^{C} \mathcal{L}_{focal}(\sum_{i\to j} \mathbf{P}_i, \mathbf{M}_j) +$$
$$\rho \sum_{j=1}^{C} \mathcal{L}_{dice}(\sum_{i\to j} \mathbf{P}_i, \mathbf{M}_j) + \gamma \sum_{i=1}^{L} \sum_{k=1}^{L} \mathcal{L}_{cos}(\mathbf{P}_{i\to j}, \mathbf{P}_{k\to j}). \quad (6)$$

Note that with a slight abuse of notation the sums in the focal and dice losses are overall concept regions i that matched to one connected component j, and are effectively modeling the union of the concept regions matched to a given component (see Figure 3 (right)). This, in effect, means that concepts are only weakly-supervised. The ρ and γ are the balancing parameters for the loss terms.

3.4. Final Loss

The final loss for our model is a combination of the traditional segmentation losses like per-pixel classification or mask-classification loss depending on what approach we choose and the latent concept loss defined above:

$$\mathcal{L} = \mathcal{L}_{seg}(\cdot, \cdot) + \beta\mathcal{L}_{concept}(\mathbf{P}, \mathbf{M}), \quad (7)$$

where $\mathcal{L}_{Seg}$ is a segmentation loss of choice (per-pixel or mask classification loss). Again, β is a balancing parameter between the two terms.

4. Metrics for interpretability

We aim to design a component that not only improves segmentation performance but have more semantic latent representations. Hence we propose a series of metrics that measure the semantics of the tokens. Our metrics rely on two core assumptions: (1) the token is semantic if its concept region belongs to a coherent object class or instance, and (2) concept regions, on the whole, should capture as many object categories and instances as possible (*i.e.*, be diverse).

Semantics ($\mathcal{S}$). To measure semantics, we first compute a histogram for each concept region. The bins of the histogram correspond to the classes present in the image. Each pixel of concept region votes for the label of its ground truth object class based on its projection weight. The histogram is then normalized to sum to 1. In practice, for each image, we compute K discrete probability distributions that measure the empirical probability of the object class belonging to the token. We measure the entropy of these probability distributions and then average the resulting K entropies. This mean entropy is used as the measure of the semantics for the tokens for a given image. Note, the *lower the entropy the more semantic the token representations are* because a lower entropy indicates a uni-modal distribution suggesting that most of the token support comes from a single object class. A dataset measure can be obtained by averaging the mean entropy over all images. We call this class-semantics $\mathcal{S}_{\mathcal{C}}$. We extend this metric to quantify the ability of our tokens to distinguish between individual object instances and not just classes and call it instance-semantics $\mathcal{S}_{\mathcal{I}}$. The metric can be computed in the same way except the bins of the histogram correspond to the individual instances for each of the "things" classes (a class that has instance annotations).

Diversity ($\mathcal{D}$) The semantics metric defined above can be minimized by having *all* tokens focus on only one object class or instance. Hence it is desirable to also measure diversity of tokens. Hence, we propose the token diversity metric. To compute the metric for an image, we first calculate the mean of the normalized histograms mentioned above and the variance of the histograms. Finally we calculate the mean of the variances for all images in the dataset. The higher the variance, the more diverse the tokens are. We desire a high token diversity. Similar to semantics, diversity can also be defined at the class- ($\mathcal{D}_C$) or instance-level ($\mathcal{D}_I$).

5. Experiments

To show the effectiveness of our framework We conduct three sets of experiments: **(1)** experiments where we demonstrate that adding SGR component multiple segmentation frameworks having different backbones and approaches (per-pixel or mask-classification based), leads to consistent improvement in performance across three benchmark datasets: Cityscapes [20], COCO-Stuffs-10k [6] and

Models	Backbone	Datasets	mIoU (s.s.)	mIoU (m.s.)
Dil-FCN* [arXiv '17] [14]	Res-101	COCO-Stuffs	37.8	38.9
GCNET (Dil-FCN + GloRE) [CVPR '19] * [17]	Res-101	COCO-Stuffs	37.1	38.3
Ours (Dil-FCN + SGR)	Res-101	COCO-Stuffs	**38.8 (+1.0)**	**39.7 (+0.8)**
Maskformer [NeurIPS '21] [19]	Res-101	COCO-Stuffs	38.0	39.3
Ours (Maskformer + SGR)	Res-101	COCO-Stuffs	**38.9 (+0.9)**	**39.9(+0.6)**
Swin-T-Upernet* [ICCV '21] [45,55]	Swin-T	COCO-Stuffs	39.1	40.0
Ours (Swin-T-UperNet + SGR)	Swin-T	COCO-Stuffs	**39.3 (+0.2)**	**40.1 (+0.1)**
Mask2former [CVPR '22] [18]	Swin-T	COCO-Stuffs	42.1	43.0
Ours (Mask2former + SGR)	Swin-T	COCO-Stuffs	**42.5 (+0.4)**	**43.3 (+0.3)**
Segformer [NeurIPS '21] [56]	MiT-B4	COCO-Stuffs	-	42.5
ProtoSeg+Segformer [CVPR '22] [70]	MiT-B4	COCO-Stuffs	-	43.3
ProtoSeg+Swin-B [CVPR '22] [70]	Swin-B	COCO-Stuffs	-	42.4
Dil-FCN* [arXiv '17] [14]	Res-101	ADE-20K	42.9	44.0
GCNET (Dil-FCN + GloRE)* [CVPR '19] [17]	Res-101	ADE-20K	43.2	44.8
Ours (Dil-FCN + SGR)	Res-101	ADE-20K	**43.8 (+0.9)**	**45.6 (+1.6)**
DeepLabV3* [arXiv '17] [14]	Res-101	ADE-20K	43.1	44.4
Ours (DeepLabV3 + SGR)	Res-101	ADE-20K	**44.9 (+1.8)**	**46.2 (+1.8)**
Swin-T-Upernet* [ICCV '21] [45,55]	Swin-T	ADE-20K	43.0	43.6
Ours (Swin-T-Upernet + SGR)	Swin-T	ADE-20K	**43.9 (+0.9)**	**45.0 (+1.4)**
ProtoSeg+ HR-Net [CVPR '22] [70]	HRNetV2-W48	ADE-20K	-	43.0
ProtoSeg+Swin-B [CVPR '22] [70]	Swin-B	ADE-20K	-	48.6
Dil-FCN * [arXiv '17] [14]	Res-101	Cityscapes-val	77.9	79.3
GCNET (Dil-FCN + GloRE)* [CVPR '19] [17]	Res-101	Cityscapes-val	78.0	79.3
Ours (Dil-FCN + SGR)	Res-101	Cityscapes-val	**78.7 (+0.7)**	**80.5 (+1.2)**
DeeplabV3* [arXiv '17] [14]	Res-101	Cityscapes-val	78.5	79.8
Ours (DeeplabV3 + SGR)	Res-101	Cityscapes-val	**79.8 (+1.3)**	**81.2 (+1.4)**
Maskformer [NeurIPS '21] [19]	Res-101	Cityscapes-val	78.5	80.3
Mask2former [CVPR '22] [18]	Res-101	Cityscapes-val	80.1	81.9
ProtoSeg+Swin-B [CVPR '22] [70]	Swin-B	Cityscapes-val	-	80.6

Table 1. **Results for semantic segmentation.** For Cityscapes and ADE-20K the results are reported on the validation set, while for COCO-Stuff-10K the results are reported on the test-set. The baseline models marked with * are trained by us under similar training settings as ours. The rest of the results are reported from the respective papers. Numbers within the parenthesis indicate improvement over the model on which SGR component is added. The mIoU (s.s.) indicates mIoU using single scale inference while mIoU (m.s.) indicate multi-scale inference.

Models	Backbone	Datasets	$\mathcal{S}_C\downarrow$	$\mathcal{D}_C\uparrow$
GCNET (Dil-FCN + GloRE) [17]	Res-101	COCO-Stuffs	0.478	0.078
Maskformer [19]	Res-101	COCO-Stuffs	0.275	0.186
Ours (Dil-FCN + SGR)	Res-101	COCO-Stuffs	**0.226**	**0.389**
Ours (Swin-UpNet + SGR)	Swin-T	COCO-Stuffs	0.227	0.376
GCNET (Dil-FCN + GloRE) [17]	Res-101	ADE-20K	0.564	0.106
Maskformer [19]	Res-101	ADE-20K	0.335	0.173
Ours (Dil-FCN + SGR)	Res-101	ADE-20K	0.264	0.391
Ours (Swin-UpNet + SGR)	Swin-T	ADE-20K	**0.240**	**0.421**
GCNET (Dil-FCN + GloRE) [17]	Res-101	Citys-val	0.741	0.055
Maskformer [19]	Res-101	Citys-val	0.413	0.209
Ours (Dil-FCN + SGR)	Res-101	Citys-val	**0.293**	**0.469**
Models	Backbone	Datasets	$\mathcal{S}_I\downarrow$	$\mathcal{D}_I\uparrow$
Maskformer [19]	Res-101	MS-COCO	0.383	0.122
Ours (Dil-FCN + SGR)	Res-101	MS-COCO	**0.315**	**0.316**
Ours (Swin-UpNet + SGR)	Swin-T	MS-COCO	0.328	0.307

Table 2. **Semantics.** Comparison of class-level ($\mathcal{S}_C$) and instance-level ($\mathcal{S}_I$) semantics of intermediate tokens and the token diversity on class ($\mathcal{D}_C$) and instance-level ($\mathcal{D}_I$). The lower the value of ($\mathcal{S}_C$) and ($\mathcal{S}_I$) the more semantic the the token representations are.

ADE-20K [69]; **(2)** experiments that compare semantics and diversity of our latent token representations with recent alternatives GCNET [17] and Maskformer [19], where we show superiority of our representations regardless of backbone or framework. Finally, **(3)** we demonstrate the effectiveness of the features learned by our component by transferring the learned weights from the semantic segmentation network and fine-tuning to downstream tasks of object detection and segmentation on the MS-COCO dataset [43]. We also performed a series of ablation studies to justify our design choices.

5.1. Datasets

Cityscapes [20] contains street-view images captured using a dashcam. It has 19 semantic classes labeled.
COCO-Stuffs-10K [6] is a subset of the MS-COCO [43]. It has pixel-level annotations of 171 semantic classes.
ADE-20K [69] is subset of the ADE20K-Full dataset containing indoor and outdoor scenes with 150 classes.
MS-COCO [43] is a large-scale dataset used for object detection, segmentation, and image captioning.

5.2. Implementation Details

Semantic Segmentation. For semantic segmentation, we add our SGR component after the final layer of the back-

Backbone	AP_{bbox}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Res101-C4 [27, 28]	40.29	59.58	43.37	22.41	44.94	55.05
Res101-GCNET [17]	38.85	58.82	42.30	21.62	42.76	51.88
Ours-Res101-SGR	**41.91**	**62.79**	**45.23**	**22.95**	**46.04**	**56.32**
Ours (w/o token sup.)	33.48	51.72	36.2	18.73	36.96	45.84
Backbone	AP_{mask}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Res101-C4	34.88	56.16	37.27	15.30	38.32	53.18
Res101-GCNET	34.35	55.08	36.43	15.07	38.33	51.37
Ours-Res101-SGR	**37.06**	**59.28**	**39.31**	**16.30**	**40.95**	**56.19**
Ours (w/o token sup.)	29.73	48.75	31.41	13.55	33.85	46.41

Table 3. **Transfer to Object Detection and Segmentation.** Transfer learning performance on object detection and instance segmentation using Mask-RCNN [27]. All the models trained on COCO **train2017** by us and evaluated on **val2017** for a fair comparison.

	COCO-Stuffs-10K			MS-COCO	
Method	mIOU	$\mathcal{S}_C\downarrow$	$\mathcal{D}_C\uparrow$	$\mathcal{S}_I\downarrow$	$\mathcal{D}_I\uparrow$
Ours	**39.7**	<u>0.226</u>	**0.389**	**0.315**	<u>0.316</u>
Ours w/o focal, dice loss and cos. sim.	38.6	0.556	0.001	0.640	0.001
Ours w/o cos. sim.	37.7	0.519	0.042	0.629	0.032
Ours w graph	39.0	0.246	0.357	0.345	0.287
Ours w/o x-y emb.	38.9	0.412	0.152	0.441	0.192
Ours w/o centr. emb.	38.6	**0.224**	0.385	**0.315**	**0.323**
Ours w/o any pos. emb.	39.2	0.453	0.106	0.496	0.143

Table 4. **Ablation Studies on COCO-Stuffs-10K.** All models for ablation studies use Res101 backbone trained on COCO-Stuff-10K. Instance semantics and instance diversity are reported on COCO val2017.

bones, pre-trained on ImageNet, just before the segmentation head. For dilated FCN and DeepLabV3 heads we use the SGD optimizer with a momentum of 0.9 [50] and a polynomial learning rate policy where the learning rate decreases with the formula $(1 - \frac{iter}{total\_iter})^{0.9}$ with an initial learning rate of 0.006 for Cityscapes and 0.004 for ADE-20K and Coco-Stuffs. For mask-classification approaches and Swin backbones, we used the AdamW optimizer with an initial learning rate of 1e-5. For Cityscapes, we use a batch size of 8 and a crop size of 768×768. For both Coco-Stuffs and ADE-20K we use a batch size of 16, crop size of 512×512. Each segmentation model was trained on two RTX6000 GPUs with a memory of 24 GB each. For all experiments, we multiply the initial learning rate by 10.0 for the parameters of the segmentation head and SGR component. We report both the single-scale inference and multi-scale inference with a horizontal flip at scales 0.5, 0.75, 1.0, 1.25, 1.5, and 1.75 following existing work [14, 19, 25, 61].

Transfer to Downstream Tasks. For transferring our model on downstream tasks, we first remove the segmentation head from the network trained on COCO-Stuffs-10K and use it as a backbone for Mask-RCNN [27] to fine-tune on MS-COCO for object detection and segmentation. We trained our model on **train2017** and evaluated on **val2017**. For a fair comparison with other backbones, we trained with the same batch size, learning rate, and iterations. We trained the models using SGD with a momentum of 0.9 and a batch size of 8 for 270K iterations. The learning rate decreased by a factor of 0.1 at 210K and 250K iterations.

Additional implementation details are in Supplemental.

5.3. Results

Semantic segmentation. The performance of our approach on semantic segmentation is given in Table 1. As observed, when our component is added on top of different segmentation frameworks, it consistently improves mIoU (mean Intersection over Union) for all three datasets for both CNN based and transformer-based backbones. It achieves an improved performance regardless of using per-pixel classification approach (like dilated-FCN or UperNet) or mask-classification approach (like Maskformer [19] and Mask2Former [18]). When we add SGR with Maskformer, there is a single scale mIOU improvement of 0.9 and multi-scale mIOU imrpovement of 0.6 on COCO-Stuffs dataset. We similarly see an improvement when we add SGR to Mask2Former, which uses multi-scale features. In fact, when compared against similar backbones/frameworks, we achieve the best result on COCO-Stuffs. Even when compared to recent methods like SegFormer and ProtoSeg, which use stronger backbones (Swin-B and MiT-B4) compared to Swin-T, our approach outperforms them.

On ADE-20K, adding our component to DeepLabV3 achieves the best performance over any approaches using Resnet101 except Maskformer. For Swin-T, although the improvement on COCO-Stuffs is marginal, the improvement on ADE-20K is significant. We observe similar improvements on the Cityscapes dataset when our component is added to DeepLabV3 and Dilated-FCN; DeepLabV3 + SGR outperforms all approaches using similar backbones except Mask2Former.

Given consistent improvements over *three* different baseline approaches that use different backbones, tested on *three* diverse datasets, we expect similar improvements to arise when SGR is added to other architectures, including variants that leverage larger backbones such as Swin-L that we were unable to train due to compute constraints.

Class and instance-semantics. We quantify the interpretability of the generated tokens for SGR at the class-level and instance-level using the metrics discussed in Section 4 and compare against intermediate representations of Maskformer [19] and tokens of GLoRE [17]. The results are shown in Table 2. As can be observed, our tokens are more semantically meaningful than other intermediate representations at both semantic and instance levels while at the same time being more diverse. We observe this for the Swin [45] backbone as well, showing the effectiveness of our connected component weak superivision.

Transfer to downstream tasks. Table 3 shows the performance of our proposed component when transferred to down-stream tasks of object detection and segmentation on MS-COCO [43] dataset using Mask-RCNN [27]. We compared against two other backbones, Resnet-101

Figure 4. **Qualitative results** showing that our SGR component generates more semantically meaningful and diverse tokens. In all three images, SGR was able to disambiguate between the instances (was able to differentiate the cow in the left in the first image, and different groups/ instances of the car in the last row) unlike Maskformer [19]. GCNET [17] tokens, on the other hand, lack strong semantic meaning.

(Res101-C4) pre-trained on Imagenet and GloRE (Res101-GCNET) [17] pre-trained on COCO-Stuffs-10K on semantic segmentation task. As can be observed from Table 3, SGR outperforms both on object detection and instance segmentation tasks. This demonstrates that the more semantically interpretable and diverse token representations allow us to learn richer features that are broadly more useful and transferable. Note that these downstream tasks require the ability to discern multiple instances and the instance-centric nature of the way our SGR aggregates information allows us to achieve this improved performance. This is further highlighted when compared against the model transferred from GloRE [17] which also aggregates features into multiple tokens and reasons between them but the tokens lack any semantic coherence. Furthermore, as can be observed from the Table 3, when we do not supervise the tokens the performance on downstream tasks drastically drops.

Ablation Studies. Table 4 highlights the importance of each of our components. As observed, when we do not supervise the tokens using mask losses, both the accuracy and semantic interpretability of tokens drop. This highlights the importance of semantically meaningful intermediate representations for better accuracy. We also observe that cosine similarity loss plays an important role in ensuring both token diversity and segmentation accuracy. Applying our losses over graph convolutions (instead of transformer encoder) allow it to have semantically meaningful tokens but the overall accuracy drops. Lastly we ablated importance of the two positional embeddings used. The x-y embedding is crucial to obtain object-centric tokens since it encodes spatial information. Similarly, the centroid embedding improves the accuracy.

Ablations for hyper-parameter sensitivity (for different values of K and L) and a discussion of computational overhead is provided in the supplementary.

Qualitative Results. Figure 4 shows qualitative results for generated tokens on multiple images. As observed, our tokens are more semantically interpretable and diverse, compared to the GCNET [17] and Maskformer [19]. Crucially, compared to Maskformer, which also supervises tokens, SGR can distinguish between instances of objects at different spatial locations; *e.g.*, in the first row, one of the tokens of SGR is able to distinguish the left cow from the other, while Maskformer [19] fails to do so. We can similarly observe that SGR is able to distinguish the rightmost horse in the second image which is disjoint from the rest and in the last image three different tokens of SGR are attending to three different groups of cars where other methods failed.

The qualitative results for semantic segmentation, object detection and segmentation are in the supplementary.

6. Conclusion

To summarize, we propose a novel component that learns to semantically group image features into latent tokens and reasons between them using self-attention. The losses we propose allow our tokens to distinguish between individual connected components of a semantic class. We also propose new metrics to demonstrate that our latent tokens are meaningful and semantically interpretable at both class- and instance-levels. Morever, we have empirically demonstrated that our component is independent of any backbone or segmentation framework and consistently improves performance when added with these approaches over three datasets and achieves the best performance over similar backbone and frameworks. Additionally, the rich set of features learned by our framework can be transferred to downstream object detection and instance segmentation tasks.

[3]**Acknowledgments and Disclosure of Funding.** This work was funded, in part, by the Vector Institute for AI, Canada CIFAR AI Chair, NSERC CRC and an NSERC DG. Hardware resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. Additional support was provided by JELF CFI grant and Compute Canada under the RAC award. Finally, we sincerely thank Gaurav Bhatt for his valuable feedback on the paper draft.

References

[1] Pablo Arbeláez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *CVPR*, pages 3378–3385. IEEE, 2012. 3

[2] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014. 3

[3] Gedas Bertasius, Lorenzo Torresani, Stella X Yu, and Jianbo Shi. Convolutional random walk networks for semantic image segmentation. In *CVPR*, pages 858–866, 2017. 1, 3

[4] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *IJCV*, 70(2):109–131, 2006. 3

[5] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004. 3

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 5, 6

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 5

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 3

[9] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443. Springer, 2012. 3

[10] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2011. 3

[11] Siddhartha Chandra, Nicolas Usunier, and Iasonas Kokkinos. Dense and low-rank gaussian crfs using deep embeddings. In *ICCV*, pages 5103–5112, 2017. 3

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 1, 3

[13] 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. *PAMI*, 40(4):834–848, 2017. 3

[14] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3, 6, 7

[15] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 3

[16] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. *NIPS*, 31, 2018. 1

[17] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, pages 433–442, 2019. 1, 3, 4, 6, 7, 8

[18] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1, 2, 3, 6, 7

[19] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NIPS*, 34:17864–17875, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[20] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5, 6

[21] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, pages 3992–4000, 2015. 3

[22] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV, pages=764–773, year=2017*. 3

[23] 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. 3

[24] Conditional Random Fields. Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001. 3

[25] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 1, 3, 7

[26] Yizeng Han, Dongchen Han, Zeyu Liu, Yulin Wang, Xuran Pan, Yifan Pu, Chao Deng, Junlan Feng, Shiji Song, and Gao Huang. Dynamic perceiver for efficient visual recognition. *ICCV*, 2023. 1

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 7

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7

[29] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990. 3

[30] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, pages 603–612, 2019. 3

[31] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 1

[32] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv preprint arXiv:2112.12782*, 2021. 1, 3

[33] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *ICCV*, pages 1780–1790, 2021. 4

[34] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 3

[35] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1

[36] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[37] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 24, 2011. 3

[38] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 5

[39] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep hierarchical semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1246–1257, 2022. 1

[40] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. *Advances in neural information processing systems*, 31, 2018. 1, 3

[41] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. *NIPS*, 31, 2018. 3

[42] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6, 7

[44] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 1

[45] 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 *ICCV*, pages 10012–10022, 2021. 3, 6, 7

[46] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 3

[47] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571. IEEE, 2016. 5

[48] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 3

[49] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 3

[50] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147. PMLR, 2013. 7

[51] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 3

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017. 2, 3, 4

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 3

[54] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph E Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Where do transformers really belong in vision models? In *ICCV*, pages 599–609, 2021. 3

[55] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 6

[56] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1, 3, 6

[57] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, pages 4310–4319, 2022. 2

[58] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, pages 4310–4319, 2022. 3

[59] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, pages 3684–3692, 2018. 1, 3

[60] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, pages 472–480, 2017. 3

[61] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, pages 173–190. Springer, 2020. 1, 2, 3, 7

[62] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Ocnet: Object context for semantic segmentation. *IJCV*, 129(8):2375–2398, 2021. 2, 3

[63] 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 *CVPR*, pages 11101–11111, 2022. 3

[64] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *ICCV*, pages 6798–6807, 2019. 1

[65] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing networks. In *CVPR*, pages 3726–3735, 2020. 3

[66] Songyang Zhang, Xuming He, and Shipeng Yan. Latentgnn: Learning efficient non-local relations for visual recognition. In *ICML*, pages 7374–7383. PMLR, 2019. 3

[67] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 3

[68] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021. 3

[69] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 6

[70] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2582–2593, 2022. 1, 6