Rethinking Self-Supervised Semantic Segmentation: Achieving End-to-End Segmentation

Yue Liu, Jun Zeng, Xingzhen Tao, and Gang Fang

***Abstract*—The challenge of semantic segmentation with scarce pixel-level annotations has induced many self-supervised works, however most of which essentially train an image encoder or a segmentation head that produces finer dense representations, and when performing segmentation inference they need to resort to supervised linear classifiers or traditional clustering. Segmentation by dataset-level clustering not only deviates the real-time and end-to-end inference practice, but also escalates the problem from segmenting per image to clustering all pixels at once, which results in downgraded performance. To remedy this issue, we propose a novel self-supervised semantic segmentation training and inferring paradigm where inferring is performed in an end-to-end manner. Specifically, based on our observations in probing dense representation by image-level self-supervised ViT, i.e. semantic inconsistency between patches and poor semantic quality in non-salient regions, we propose *prototype-image alignment* and *global-local alignment* with *attention map constraint* to train a tailored Transformer Decoder with learnable prototypes and utilize adaptive prototypes for segmentation inference per image. Extensive experiments under fully unsupervised semantic segmentation settings demonstrate the superior performance and the generalizability of our proposed method.**

***Index Terms*—Self-supervised learning, semantic segmentation.**

I. Introduction

UNDERSTANDING image at pixel-level granularity still stands as one of the most important and challenging tasks in computer vision due to the following proverbial facts: extreme scarcity of pixel-level annotations and non-trivial effort transferring knowledge learned from coarse granularity to finer granularity (e.g. image-level to pixel-level granularity) [1]. Recently, the vision community resorts to self-supervised learning for novel solutions of pixel-level tasks, e.g. semantic segmentation.

Manuscript received 13 October 2023; revised 29 April 2024; accepted 18 July 2024. Date of publication 23 July 2024; date of current version 5 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61972107 and in part by the National Key R and D Program of China under Grant 2019YFA0706338402. Recommended for acceptance by C. Wolf. *(Corresponding author: Gang Fang.)*

Yue Liu is with the Institute of Computing Science and Technology, Guangzhou University, Guangzhou 511370, China, and also with the School of Information Engineering, Jiangxi College of Applied Technology, Ganzhou 341003, China (e-mail: liuyue1229@qq.com).

Jun Zeng and Xingzhen Tao are with the School of Information Engineering, Jiangxi College of Applied Technology, Ganzhou 341003, China.

Gang Fang is with the Institute of Computing Science and Technology, Guangzhou University, Guangzhou 511370, China (e-mail: gangf@gzhu.edu.cn).

The code is available at: https://github.com/yliu1229/AlignSeg

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3432326, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3432326

Without pixel-level annotations, many works introduce domain-specific priors to assist self-supervised dense representation learning, which include contour detectors [2], [3], saliency detectors [4], [5], region proposals [6], [7], or clustering [8], [9]. However, the dense representation learning is largely limited by these hand-crafted priors consequentially. Recently, image-level self-supervised ViTs are found to be able to produce semantic information in dense representations [10], therefore facilitating many self-supervised semantic segmentation works like STEGO [11] and ACSeg [12] that preliminarily explore dense prediction with fixed pre-trained models. Some other works design dedicated self-supervised pre-training tasks for dense representation learning [13], [14], and achieve impressive performance in downstream semantic segmentation tasks.

Nevertheless, most self-supervised semantic segmentation methods essentially train an image encoder or a segmentation head that produces finer dense representations, and when performing semantic segmentation inference or evaluation, they still need to resort to fine-tuned linear classifier or traditional clustering (e.g. K-means). We illustrate typical evaluation protocols of self-supervised semantic segmentation methods in Fig. 1. Fig. 1(a) fixes the self-supervised encoder and fine-tunes a linear classifier with pixel-level annotations for segmentation results, which essentially makes this evaluation supervised. Under fully unsupervised settings, final segmentation results are obtained by clustering dense representations on dataset level as Fig. 1(b). The inference practices in both Fig. 1(a)(b) significantly deviate from the conventional semantic segmentation where segmentation results are inferred in a real-time and end-to-end manner. More importantly, clustering on dataset level or at least large-batch level actually escalates the problem from segmenting per image to clustering all pixels at once, which we argue is one key aspect that the evaluation performance of unsupervised semantic segmentation methods greatly lag behind supervised methods. On the one hand, clustering is neither elegant nor efficient and often yields inconsistent evaluation results [15]. On the other hand, without pixel-level categorical supervision by ground-truth annotations, the dense representations learned by self-supervision may only contain coarse categorical information which causes difficulty in categorical alignment in dataset-level embedding space, thus clustering or even linear probe may result in downgraded performance (fine-tuning a linear classifier is essentially clustering dataset-level dense representations to semantic classes in a supervised manner), we will show more about this observation in Section III. To remedy this issue, we propose a novel self-supervised semantic segmentation training

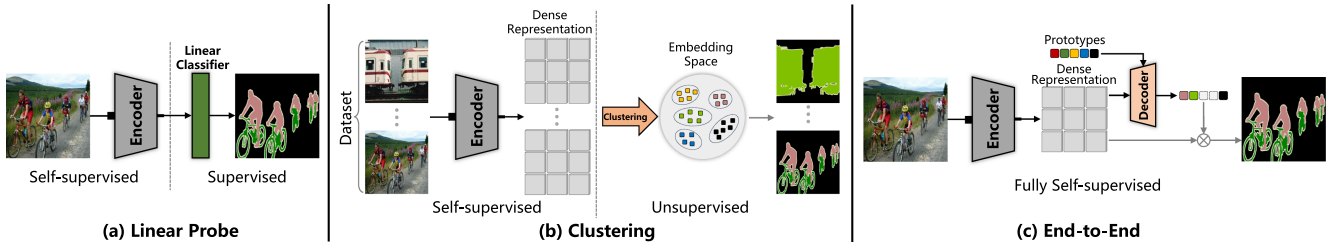

Fig. 1. Evaluation protocols of self-supervised semantic segmentation methods.

and inferring paradigm as shown in Fig. 1(c) where inferring is performed in a real-time and end-to-end manner. Instead of traditional clustering, we introduce prototypes that are extracted per image adaptively and then used for inference by a simple similarity $argmax$ function.

Following prior works [11], [12], we break down self-supervised semantic segmentation into self-supervised dense representation learning and self-supervised segmentation head training, in this work, we primarily focus on the latter based on fixed self-supervised image encoders. Although in this paper we only use image-level self-supervised ViT (e.g. DINO [10]) to illustrate our self-supervised training method, the proposed method can be flexibly integrated with other self-supervised dense representation learning frameworks to achieve real-time and end-to-end semantic segmentation. Inspired by DETR [16] and MaskFormer [17], we utilize learnable prototypes to query image dense representations through a tailored Transformer Decoder [18]. VITA [19] also explores a token association mechanism for video instance segmentation, where an object detector is used to distill object-specific contexts into object tokens which are similar to the concept of prototypes in our work. And based on our observations in probing dense representations by self-supervised ViT, we propose two *Align*ment methods for self-supervised semantic *Seg*mentation training (AlignSeg), we elaborate our observations in Section III and the alignment methods in Section IV.

We extensively evaluate the proposed AlignSeg under fully unsupervised semantic segmentation settings on commonly used datasets, including PASCAL VOC 2012 [20] and COCO-Stuff [21]. Thanks to the new inferring paradigm as in Fig. 1(c), we can evaluate semantic segmentation end-to-end and image-by-image without fine-tuning extra components or running extra clustering. We also show that the proposed AlignSeg can be used as a generalizable method that is easily integrated with current self-supervised dense representation frameworks.

To summarize, our main contributions are as follows:

- We propose a novel self-supervised semantic segmentation training and inferring paradigm where inferring is performed in a real-time and end-to-end manner. We utilize *prototype-image alignment* and *global-local alignment* with *attention map constraint* to train a tailored Transformer Decoder with learnable prototypes and then use adaptive prototypes for segmentation inference per image.
- We probe the dense representations produced by image-level self-supervised ViTs and reveal the semantic inconsistency between patches and the poor semantic quality in non-salient regions, and prove unsupervised segmentation inference by clustering results in downgraded performance.
- We demonstrate that under fully unsupervised semantic segmentation settings, the proposed method achieves state-of-the-art segmentation performances.

II. Related Works

A. Self-Supervised Learning

Self-supervised learning, especially image-level representation learning, has been extensively studied in the vision community. Self-supervised learning methods mainly utilize two types of optimization target, i.e. contrastive target [22], [23] and generative target [24], [25], to learn visual representations which benefit downstream tasks. Recently, self-distilled method DINO observes image-level self-supervised ViT outputs semantic-aware dense representations which leads to the emergence of object segmentation in the attention map [10].

Despite the success of self-supervised learning in image-level downstream tasks (e.g. classification, retrieval), it still faces difficulties in transferring to pixel-level tasks (e.g. object detection, semantic segmentation), thus motivating some self-supervised dense representation learning works which design dedicated self-supervised pixel-level pre-training tasks [14], [26], [27], [28], [29]. These works utilize pixel-level pre-training objectives that facilitate the learning of finer dense representations, and they indeed achieve better performance in downstream pixel-level tasks. However, these methods are not specially designed for semantic segmentation and normally need additional supervision when transferring to segmentation tasks.

B. Self-Supervised Semantic Segmentation

Aiming at semantic segmentation, many self-supervised dense representation learning works introduce domain-specific priors, including contour detectors [2], [3], saliency detectors [4], [5], region proposals [6], [7], or clustering [8], [9]. However, the dense representation learning is largely limited by these hand-crafted priors consequentially. Recently, inspired by semantic-aware dense representations in DINO [10], self-supervised semantic segmentation works like Leopart [13] and HP [15] use image-level pre-trained models as initialization and guidance, works like STEGO [11] and ACSeg [12] further explore dense prediction with fixed pre-trained models, other works utilize image-level self-supervised ViTs in various

ways [30], [31], [32], [33], [34]. This stream of works shows that data-driven self-supervised models provide stronger prior knowledge for semantic segmentation training than those hand-crafted priors, and decoupling self-supervised dense representation learning and self-supervised semantic segmentation training gives promising results. We also extend this stream of works and train segmentation head upon fixed self-supervised models.

However, most self-supervised semantic segmentation methods essentially train an image encoder or a segmentation head that produces finer dense representations, and when performing semantic segmentation inference or evaluation, they still need to resort to fine-tuned linear classifier or traditional clustering (e.g. K-means), as illustrated in Fig. 1(a) and (b). MaskDistill [34] facilitates a multistage self-supervised training framework based on its hand-made rules and realizes end-to-end semantic segmentation inference, but it is complex and comprises training of several models. Although these self-supervised semantic segmentation methods can fine-tune a segmentation head using pseudo-labels generated by clustering as in LUSS [35], the evaluation is essentially the same as clustering, and the segmentation quality using noisy clustering labels is poor. To remedy this issue, we propose a novel self-supervised semantic segmentation training and inferring paradigm where inferring is performed in a real-time and end-to-end manner.

Recently, with the prevalence of vision-language pre-training methods, works like GroupViT [36] and SegCLIP [37] show that object segments automatically emerge with text supervision, they also utilize learnable centers to segment semantic regions.

III. Probing Image-Level Self-Supervised ViT

Image-level self-supervised ViTs are proven to be able to output semantic-aware dense representations, but the semantic quality is unclear. Hence, it is crucial to probe these dense representations thoroughly if we are to build a segmentation module upon frozen pre-trained ViTs.

A. Semantic Consistency Between Patches

DINO [10] shows the attention map of [CLS] token contains segmentation information, indicating that the patch embedding contains semantic information. We find this property exists in other image-level self-supervised models as well, such as MoCo [23] and even generative method MAE [24]. In terms of semantic segmentation, it is important to know how much semantic information patches contain and how well the semantic features are aligned between patches. Generally, the better semantic consistency produced by ViT, the better semantic segmentation performance we will get.

We introduce the concept of prototype, which can be interpreted as the representative of a semantic class, to probe semantic consistency. We average the embedding of patches highlighted by the attention map as a semantic prototype, and use the prototype to segment the same semantic patches of different images with similarity function. We use the pre-trained ViT-S/16 and threshold the averaged attention map as in DINO, then we extract the semantic prototype and use it to segment the same semantic patches by thresholding the similarity map which is computed by cosine similarity between the prototype and patch representations, similar to how we process the attention map. Results are shown in Fig. 2, where the attention map and similarity map are thresholded using the hand-picked best ratio, i.e. the ratio is selected by manually checking the best IoU between the binary mask and the ground-truth semantic region when adjusting the threshold by 5% each time until a best ratio is selected. We also visualize the patch embedding space with ground-truth class annotations in different colors. Notably, although the patch embeddings of dogs seem to be well clustered, as the number and variance of patches increase for the same semantic class, patch embeddings start to scatter and to mix with other patches in the embedding space, which is lethal for semantic categorization.

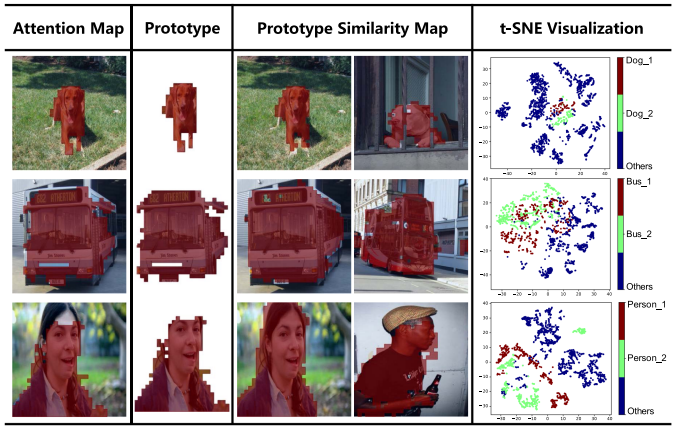

Fig. 2. Probing semantic consistency between patches: "Attention Map" column shows the [CLS] token's attention map after thresholding as DINO, then the semantic prototype is extracted according to the attention map; "Prototype Similarity Map" column shows the similarity map after thresholding, which is computed by cosine similarity between the prototype and the patches; the rightmost column shows the visualization of the patch embedding space with ground-truth class annotations in different colors.

Probing semantic consistency between patches reveals that without pixel-level categorical supervision by ground-truth annotation, variance exists between same semantic patch embeddings in a single image, and becomes non-negligible when scaling up to multiple images. IPMT [38] introduces this semantic inconsistency as intra-class diversity, I2F [39] and GMMSeg [40] suggest to perform feature adaption for semantic segmentation. Thus, it is undoubted that unsupervised segmentation by traditional clustering (e.g. K-means) on dataset level would only get downgraded performances. To tackle this problem, we propose to adaptively extract prototypes by a tailored Transformer Decoder (i.e. *prototype-image alignment*) and perform segmentation inference per image instead of on dataset level, details are described in Section IV.

B. Semantic Quality Across Image

Good semantic segmentation performance requires good semantic quality across the whole image, even in small object or edge regions. To probe the semantic quality provided by self-supervised ViT, we utilize two hand-picked prototypes (for

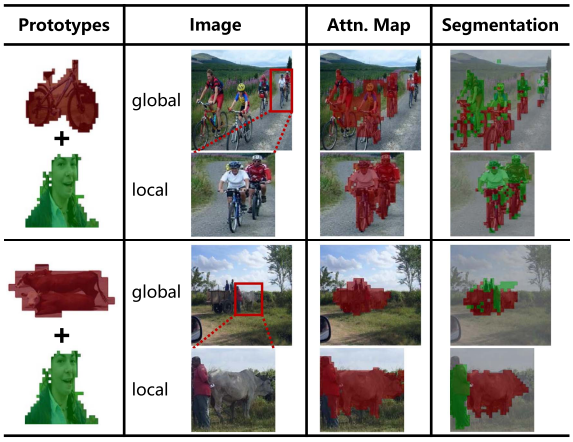

Fig. 3. Probing semantic quality across image: "Prototype" column indicates the semantic prototypes used; "Image" column shows the global and local crop of an image; "Attn. Map" column shows the attention map for both crops after thresholding; and the rightmost column shows the segmentation result with each mask being colored as same as the corresponding prototype, and the grey mask representing the background. Best viewed with zoom-in.

a given image, we manually identify two main semantic classes and select two representative images each containing one semantic class from the dataset, and then the two representative prototypes are obtained using the same method as we use in Fig. 2) to compute similarity matrix with patches and then segment regions by $argmax$ function. We show typical results in Fig. 3, where each segmentation mask is colored as same as the corresponding prototype, and the grey mask represents the background. Background is segmented using a prototype extracted by the reserve mask of the person image's attention map.

Besides the semantic inconsistency issue aforementioned, in Fig. 3 it is noticeable that the non-salient image regions, such as small objects and edge regions, are poorly segmented. We believe it is due to the poor semantic quality in these regions. Interestingly, when re-segmenting the cropped local regions, finer segmentation is normally obtained, although the refinement is limited for small objects due to limited pixel information. STEGO [11] and Leopart [13] also prove that learning from multiple global and local crops improves dense prediction performance. Furthermore, we find that the attention map can serve as a saliency mask for screening image patches, it is also utilized as foreground hint in [12], [13].

Based on the observations, in order to make the best of self-supervised ViT to train a segmentation module, we propose *global-local alignment* and *attention map constraint*, details are described in Section IV.

IV. Method

Here we elaborate the proposed AlignSeg framework, including the overall structure, the self-supervised training objective and the end-to-end semantic segmentation inference process. AlignSeg is built upon a frozen self-supervised image encoder and is trained in a fully self-supervised fashion without any annotations.

A. Segmentation Module

As illustrated in Fig. 4, AlignSeg consists of a frozen ViT $f(\cdot)$ and a tailored Transformer Decoder $Dec()$ with a set of K learnable prototypes $P \in \mathbb{R}^{K\times D}$. For an image, ViT outputs dense representations $x \in \mathbb{R}^{n\times D}$ where n is the number of patches of the image, and a binary mask $m \in \mathbb{R}^{n\times 1}$ is obtained by thresholding 60% of the attention map, which is computed exactly the same as DINO [10].

Inspired by Mask2Former [41] (a brief introduction of the Mask2Former architecture can be found in Appendix A, available online), we construct our Transformer Decoder with L layers, each of which consists of a cross-attention operation, a self-attention operation and a feed-forward network (FFN). Decoder takes prototypes P as query and dense representations x as key and value, and outputs the image-aligned prototypes $P^x \in \mathbb{R}^{K\times D}$. We refer readers to the original work of Transformer [18] for more details of the cross-attention, self-attention and FFN operations. Here we define the function of querying image dense representations x with prototypes P through our Transformer Decoder as follows:

$$P^x = Dec(P, x) \tag{1}$$

After extracting prototypes P^x adaptively per image, we compute the cosine similarity matrix s between prototypes and image patches as follows:

$$s = \cos < x, P^x > \tag{2}$$

where $s \in \mathbb{R}^{n\times K}$. Segmentation results are simply obtained by the prototypical assignment of each patch with $argmax$ function:

$$\hat{y} = \arg\max_{K}(s) \tag{3}$$

where $\hat{y} \in \mathbb{R}^{n\times 1}$ and for a patch i, we have $\hat{y}_i \in [0, K)$.

B. Self-Supervised Segmentation Training

In Section III we show that semantic segmentation can be simply achieved by computing similarity between hand-picked prototypes and patches, however the semantic inconsistency and poor semantic quality in non-salient regions make the segmentation performance less attractive, besides, we would want AlignSeg to extract prototypes automatically. Therefore, we propose two alignment methods, i.e. *prototype-image alignment* and *global-local alignment* with *attention map constraint*, to train a tailored Transformer Decoder and a set of learnable prototypes in a self-supervised fashion.

Prototype-image alignment: We randomly initialize the learnable prototypes P and extract image-aligned prototypes P^x for each image crop as described in Section IV-A. The extracted prototypes P^x should be distinct and represent different semantic classes, while the i-th prototype P^x_i for each crop should be semantically consistent. Thus we utilize a classic contrastive loss InfoNCE [22] to constrain prototypes extracted for different

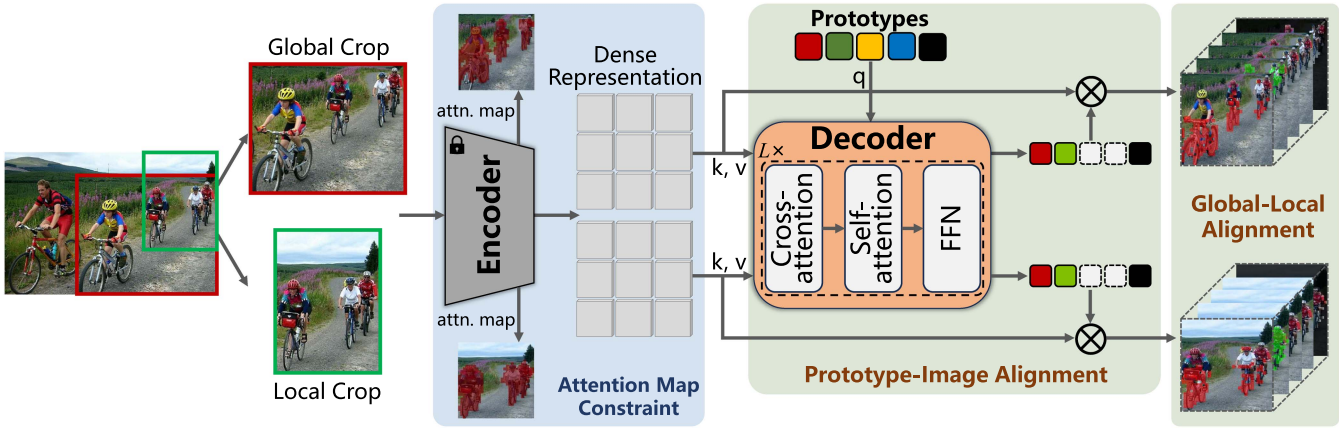

Fig. 4. Overview of the AlignSeg framework.

crops as follows:

$$\mathcal{L}_{proto} = -\frac{1}{|v_l|}\sum_{v}\sum_{i}\log\frac{\exp\left(\cos< P_i^{x_g}, P_i^{x_l^v} > /\tau\right)}{\sum_j^K \exp\left(\cos< P_i^{x_g}, P_j^{x_l^v} > /\tau\right)} \tag{4}$$

assuming we have one global crop and v_l local crops for an image, and $P_i^{x_g}$ denotes i-th extracted prototype for the global crop, $P_i^{x_l^v}$ denotes the i-th extracted prototype for the v-th local crop where $v \in \{1, 2, \ldots, v_l\}$, τ is the temperature parameter to control the softness.

Global-local alignment: Due to the nature of patch-based encoding and the limit of self-supervised ViT, non-salient image regions are poorly handled. Training with multiple local crops enable the model to look at closer details of different regions and to refine segmentation performance.

For an image x, we first randomly crop it into one global view and v_l local views, with the constraint that there should be a minimum intersection in each global-local pair. We enforce this constraint such that there would always exist strong alignment signal for global-local segmentation pairs. For a global crop, we get its dense representations $x_g \in \mathbb{R}^{n_g \times D}$ and extract its prototype $P^{x_g} \in \mathbb{R}^{K \times D}$ with (1), then compute the prototypical similarity matrix $s_g \in \mathbb{R}^{n_g \times K}$ with (2); similarly, we get $x_l^v \in \mathbb{R}^{n_l^v \times D}$, $P_i^{x_l^v} \in \mathbb{R}^{K \times D}$ and $s_l^v \in \mathbb{R}^{n_l^v \times K}$ for the v-th local crop. We then define the global-local prototypical similarity correlation:

$$ProtoCorr_{g,l_v} = s_g \left(s_l^v\right)^T \tag{5}$$

where $ProtoCorr_{g,l_v} \in \mathbb{R}^{n_g \times n_l^v}$. Intuitively, element of $ProtoCorr_{g,l_v}$ should be large if global patch and local patch are semantically correlated and small if their semantic features do not correlate. Thus, we exploit the corresponding global-local dense representation correlation as the optimization target of $ProtoCorr_{g,l_v}$. The dense representation correlation is computed as:

$$DenseCorr_{g,l_v} = \cos < x_g, x_l^v > \tag{6}$$

where $DenseCorr_{g,l_v} \in \mathbb{R}^{n_g \times n_l^v}$. We then define the *global-local alignment* objective with a simple element-wise multiplication of $ProtoCorr_{g,l_v}$ and $DenseCorr_{g,l_v}$:

$$\mathcal{L}_{corr} = -\frac{1}{|v_l|}\sum_{v} ProtoCorr_{g,l_v} \odot (DenseCorr_{g,l_v} - h) \tag{7}$$

where h is a hyper-parameter that adds a negative pressure. Optimizing $\mathcal{L}_{corr}$ with respect to $DenseCorr_{g,l_v}$ trains the Segmentation Module to produce global segmentation result that is semantically aligned with local segmentation results.

In experiments, we find the training is unstable especially when optimizing $ProtoCorr_{g,l_v}$ for weakly correlated or uncorrelated patches, thus we adopt the "0-Clamp" modification as [11], [12] and update $ProtoCorr_{g,l_v}$ as:

$$ProtoCorr_{g,l_v} = \bar{s}_g \bar{s}_l^v, \ \bar{s} = \max(s, 0) \tag{8}$$

By introducing h and "0-Clamp", we make the training focus more on strong alignment signal while mitigating the instability brought by ambiguous alignment signals.

Attention map constraint: When probing semantic quality across image, we find that the semantic quality of non-salient region is poor which would jeopardize the segmentation training. Although "0-Clamp" mitigates the issue to some extend, we further utilize the binary mask m from the attention map which can serve as a saliency mask to stabilize and accelerate the training.

We first extract the binary mask $m_g \in \mathbb{R}^{n_g \times 1}$ and $m_l^v \in \mathbb{R}^{n_l^v \times 1}$ for the global and the v-th local crop respectively, then we compute the saliency intersection mask as:

$$\mathcal{M}_{g,l_v} = \hat{m}_{g,n_l^v} \odot \left(\hat{m}_{l,n_g}^v\right)^T, \ \hat{m}_{x,n} = m_x.repeat(n) \tag{9}$$

where $repeat(n)$ denotes repeating the tensor along its first dimension by n times. Then we constrain the global-local optimization within the saliency intersection region by updating the *global-local alignment* objective as follows:

$$\mathcal{L}_{corr} = -\frac{1}{|v_l|}\sum_{v} \mathcal{M}_{g,l_v} \odot ProtoCorr_{g,l_v}$$

$$\odot(DenseCorr_{g,l_v} - h) \tag{10}$$

Overall optimization target: Combining *prototype-image alignment* as (4) and *global-local alignment* with *attention map constraint* as (10), the final self-supervised training objective is defined as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{proto} + \beta\mathcal{L}_{corr} \tag{11}$$

where α and β are coefficients that control the training stability and the diversity of prototypes.

C. End-to-End Segmentation Inference

As shown in Fig. 1(c), after self-supervised training, we get a Transformer Decoder $Dec()$ and a set of prototypes P that are dataset specific. For an image, it is first resized to (H, W) and sent to the ViT to get its dense representations x. We then query x with prototypes P through $Dec()$ to get the image specific prototypes P^x as (1), and compute the cosine similarity matrix s as (2). The final segmentation results $\hat{y}$ are obtained by the prototypical assignment of image patches as (3). As the model is trained in a fully self-supervised manner, the Segmentation Module can not assign semantic labels for the prototypical segmentation masks. For evaluation and visualization purposes, as in prior art, we align the prototypes P^x and the ground-truth semantic labels using Hungarian Matching [42].

V. Experiments

We evaluate the proposed AlignSeg on standard segmentation datasets and compare with state-of-the-art self-supervised methods. We then ablate different design choices, and extend AlignSeg to various self-supervised image encoders. We further probe the prototypes to explore its adaptability. Implementation details are provided in Appendix B, available online.

A. Setup

For fair comparison, we train a baseline AlignSeg model with one layer Decoder and five prototypes (i.e. $L = 1, K = 5$), using the image-level self-supervised ViT-S/16 from DINO [10] in experiments unless otherwise stated, the parameters of ViT are frozen during AlignSeg training. The segmentation results from AlignSeg are directly used for evaluation, without any post-processing, e.g. Conditional Random Field (CRF) [45] or K-means clustering. We report results in mean Intersection over Union (mIoU).

Dataset: We evaluate AlignSeg on two commonly used semantic segmentation datasets: the PASCAL VOC 2012 (PVOC) [20] and COCO-Stuff [21]. Following prior works [9], [11], [12], [15], [43], we use the the 27-classes version of COCO-Stuff with the "curated" split introduced by IIC [43].

Evaluation protocol: As seen in Fig. 1(c), we use the proposed self-supervised semantic segmentation inferring paradigm for evaluation, where the segmentation result is directly outputted and evaluated for each image. Without fine-tuning extra classifier or running extra clustering, AlignSeg is evaluated in a fully unsupervised manner. Thus the two previously-adopted evaluation methods, i.e. linear probe and clustering, are discarded.

TABLE I
State-of-The-Art Comparison on Pascal VOC 2012 and COCO-Stuff-27

Method	Encoder	Frozen	Evaluation Method	PVOC	COCO Stuff27
IIC [43]	ResNet18	×	Clustering	9.8	6.7
PiCIE+H [9]	ResNet18	×	Clustering	-	14.4
MaskContrast [4]	ResNet50	×	Clustering	35.0	8.9
MaskDistill [34]	ResNet50	×	End-to-End	45.8	-
DINO [10]	ViT-S/16	×	Clustering	4.6	9.6
DINO [10]	ViT-S/16	×	Linear Probe	50.6	29.4
Leopart [13]	ViT-S/16	×	Clustering	41.7	-
STEGO [11]	ViT-S/16	✓	Clustering	-	23.7
DINOSAUR [44]	ViT-B/16	✓	Clustering	37.2	24.0
ACSeg [12]	ViT-S/16	✓	Clustering	47.1	16.4
AlignSeg	ViT-S/16	✓	End-to-End	**69.5**	**35.1**

"Frozen" column indicates whether the encoder is frozen(✓) or not(×) during training. Results are reported in mean intersection over union (mIoU).

Please note since the prototypical segmentation mask is class-agnostic, we match the extracted prototypes to the ground-truth classes using Hungarian Matching [42] for evaluation, details can be found in Appendix, available online.

B. Results

Quantitative Results: We report our main results in Table I, most of the results are brought from [11], [12], and note that both "Clustering" and "End-to-End" are fully unsupervised evaluation methods while "Linear Probe" is not. AlignSeg significantly outperforms prior state-of-the-art methods. Specifically, under unsupervised evaluation settings, AlignSeg improves by 22.4 mIoU on PVOC and 11.4 mIoU on COCO-Stuff-27, compared to the next best method. We attribute the significant improvements to the end-to-end training and inferring paradigm, which allows segmentation results to be evaluated image by image and extricates from the compromised clustering evaluation. Surprisingly, AlignSeg even outperforms DINO under linear probe, as fine-tuning a linear classifier is essentially clustering dataset-level dense representations to semantic classes in a supervised manner. It further proves that unsupervised or self-supervised methods can not guarantee perfect semantic alignment of dense representations without pixel-level categorical supervision by ground-truth annotations, as suggested in Section III. In contrast, AlignSeg does not perform segmentation inference on dataset level, but focuses on segmenting the current image based on semantic difference of dense representations, achieving adaptability to different images. MaskDistill [34] also performs end-to-end segmentation inference by a DeepLab-v3 model [46] which is trained with two self-supervised learning stages, in which errors could be accumulated without ground-truth guidance. Thus, although MaskDistill infers like supervised segmentation methods, it does not provide corresponding performance.

Qualitative Results: We visualize the qualitative results on PVOC and COCO-Stuff-27 in Figs. 5 and 6 respectively, segmentation masks are not post-processed and are marked with

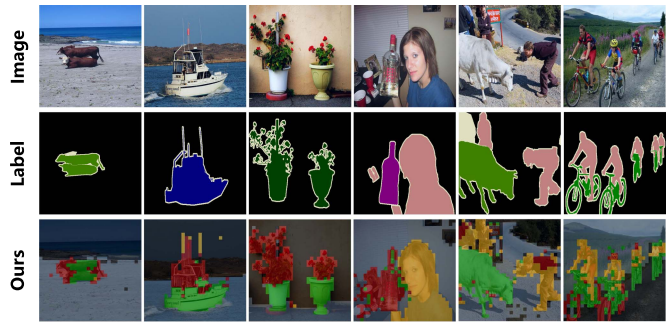

Fig. 5. Qualitative results on Pascal VOC 2012.

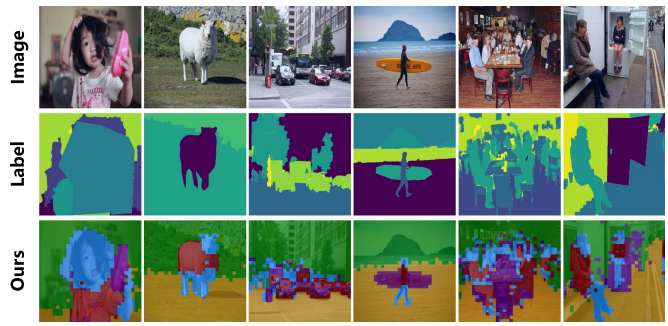

Fig. 6. Qualitative results on COCO-Stuff-27. Best viewed with zoom-in.

corresponding colors for different prototypes. In Fig. 5, it is obvious that AlignSeg performs especially well for object-centric images in PVOC, as the image encoder is pre-trained using the same type of images. AlignSeg also handles scene-centric images successfully with five prototypes capturing different semantics. It is noted that the prototypes adapt to different semantics for each image, while the semantic consistency of prototypes is preserved across different images. For example, segmentation colored in green tends to be the body of something, person segmentation is colored in yellow, and two of the prototypes always select background regions which are colored in black and gray. We also find that for images containing only one major object (e.g. 1st and 2nd column in Fig. 5), AlignSeg tends to segment object parts as Leopart [13], and for images with less semantic diversity (e.g. 1st and 3 rd column in Fig. 5), not all prototypes have to segment a region. Similar results are observed in Fig. 6, but COCO-Stuff-27 contains complex scene-centric images mostly. AlignSeg manages to segment the major semantic regions for images of different scenes, e.g. indoor, outdoor and street. However, limited by the self-supervised image encoder and the number of prototypes, the segmentation results are relatively coarse. Please note that in Figs. 5 and 6, as we do not apply any post-processing for segmentation masks, the prototypical segmentation of patches are directly used for visualization. We show examples of segmentation masks post-processed by interpolation in Appendix C, available online.

In general, AlignSeg performs better for object-centric images as the image encoder is pre-trained by image-level task on ImageNet, and it is obvious that non-salient regions are handled worse than salient regions. It should be aware that unlike SAM [47], segmentation results of AlignSeg are semantically consistent, i.e. objects of same semantic are segmented as one segmentation mask. We show additional qualitative results and failure cases in Appendix C, available online.

C. Ablation Study and Analysis

Here we ablate different design choices, including the construction of loss function, the number of Decoder layers, prototypes and crops. Furthermore, we analyze the training efficiency, and more importantly, we analyze the effectiveness and adaptability by extending AlignSeg to various self-supervised image encoders. Ablations are performed on PVOC *val* unless otherwise stated.

Design and training choices: The results of ablation study on major design choices, i.e. the number of Decoder layers L, the number of learnable prototypes K, and the number of local crops v_l, are shown in Table II(a). The proposed Decoder is quite efficient with only one layer (i.e. $L = 1$), and when using three layers, the performance improvement is marginal. Learning from multiple crops is important, when $v_l = 2$ the performance improves by 2.6, but exploiting more crops does not help anymore. Another important design is the number of prototypes, more prototypes generally results in segmentation with finer granularity, which contributes to the performance gain when using many-to-one matching as [13], [43]. However, it should be noted that the encoding capability of image-level self-supervised ViT is limited, non-salient image regions may be overlooked, thus it is impracticable to use too many prototypes because the diversity and explainability of prototypes cannot be guaranteed as one prototype may only represent a small portion of an object part and easily get confused with other prototypes. We also ablate the construction of loss function, i.e. $\mathcal{L}_{proto}$. Utilizing the prototype constraint gives a notable improvement as shown in Table II(b), we also observe that the segmentation masks are smoother. With regard to training efficiency, Table II(c) shows that AlignSeg can be trained quite efficiently with only 10 epoches and it does not require large batch size, which makes AlignSeg training feasible on a single GPU. Furthermore, we note that training on large scene-centric dataset improves the segmentation performance in general as shown in Table II(d).

Extending to various self-supervised methods: We analyze the effectiveness and adaptability of AlignSeg by extending the method to various self-supervised image encoders, results are shown in Table III. We first test AlignSeg with image-level self-supervised methods, e.g. DINO [10] and MoCo-v3 [23], and then extend it to pixel-level self-supervised semantic segmentation methods, e.g. Leopart [13] and HP [15], which are further pre-trained on COCO-Stuff. The baseline performances are evaluated by clustering under unsupervised evaluation settings (as shown in Fig. 1(b)), and most of the baseline results are taken from [13]. According to Table III, we observe the following four insights. First, AlignSeg successfully adapts to various self-supervised methods and the segmentation performance improves with the quality of the encoder. Second, it is

TABLE II
ABLATION STUDY ON DIFFERENT DESIGN AND TRAINING CHOICES

Layers	Prototypes	Crops	mIoU
1	5	2	69.5
3	5	2	69.6
1	5	1	66.9
1	5	3	69.5
1	10	2	73.2
1	15	2	73.3

(a) Design choices ablation

$\mathcal{L}_{corr}$	$\mathcal{L}_{proto}$	mIoU
✓	✓	69.5
✓	×	66.4
×	✓	29.5

(b) Loss ablation

Batch	Epoch	mIoU
32	10	69.5
32	50	69.8
128	10	68.0

(c) Training efficiency

Train	Eval.	mIoU
PVOC	PVOC	69.5
	COCO	32.5
COCO	PVOC	70.7
	COCO	35.1

(d) Training dataset

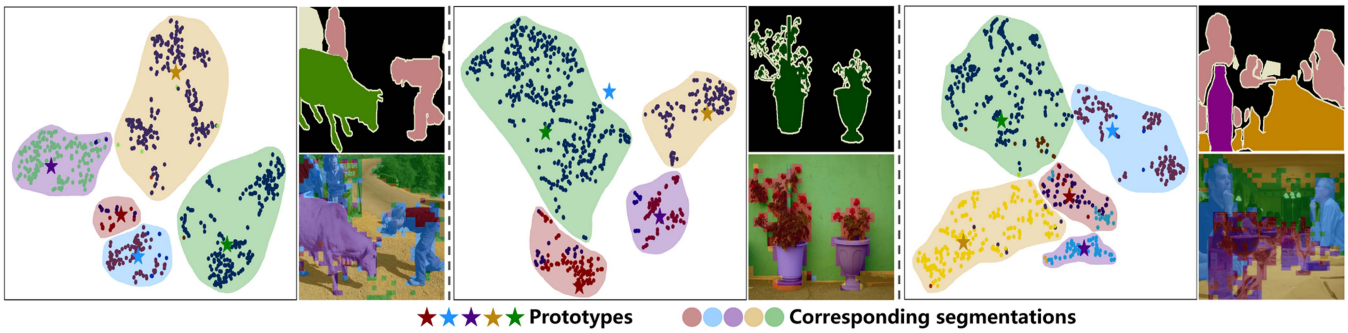

Fig. 7. t-SNE visualization of the patch embeddings and the extracted prototypes. Patch embeddings are colored according to their ground-truth annotations, and are assigned to corresponding prototypical segmentation regions.

TABLE III
EXTENDING ALIGNSEG TO VARIOUS SELF-SUPERVISED METHODS

Method	Dataset	Encoder	Pixel-level	mIoU	Impv.
DINO [10]	ImageNet	ViT-S/16	×	69.5	+64.9
DINO [10]	ImageNet	ViT-B/16	×	70.4	+65.1
MoCo-v3 [23]	ImageNet	ViT-S/16	×	52.1	+38.7
SwAV [10]	ImageNet	ResNet-50	×	56.7	+43.0
MAE [24]	ImageNet	ViT-B/16	×	64.6	+46.1
DINOv2 [48]	LVD-142M	ViT-S/14	×	72.3	+39.3
Leopart [13]	COCO-Stuff	ViT-S/16	✓	71.2	+29.5
HP [15]	COCO-Stuff	ViT-S/8	✓	71.6	+23.1

"Method" column indicates the self-supervised method to be integrated with AlignSeg, "Dataset" column shows the dataset for the "Encoder" pre-training. The segmentation performance after integrating with AlignSeg on PVOC is shown in "mIoU", and "Impv." column shows the performance improvement compared with the self-supervised baseline under unsupervised evaluation settings.

apparent that generally dedicated self-supervised dense representation learning methods output patch embeddings with better semantic quality, which improves the segmentation performance when integrating with AlignSeg. Third, AlignSeg is adaptive to both contrastive and generative self-supervised methods, and is also adaptive to different encoder architectures, e.g. ViT and ResNet. Fourth, it is surprising that although DINOv2 [48] is not specifically pre-trained for semantic segmentation, it still outperforms some self-supervised semantic segmentation methods (e.g. Leopart [13] and HP [15]) when integrating with our method.

D. Probing the Prototypes

Here we probe the prototypes to explore its adaptability and to prove the effectiveness of the proposed Transformer Decoder, in particular we analyze the extracted prototypes P^x.

Generalizability and consistency: We visualize segmentation results of different types of objects in Fig. 8. It is obvious that the five prototypes are semantically generalizable and consistent across different types of objects, e.g. furniture, person, animal, boat and plant. For example, "Proto-1" (1st column) tends to segment the head of an object or something on top while "Proto-3" always extracts the object body, and both "Proto-4" and "Proto-5" are backgrounds. It is interesting that "Proto-2" does not always segment a region (e.g. 2nd column in 1st, 3rd and 4th row), but it is sensitive to head and legs of person, we believe "Proto-2" is trained to be responsible for "Person" which is the most common object in PVOC dataset. Fig. 8 shows the prototypes successfully learn the most frequently occurring semantic concepts and can well adapt to different image content. It also proves that the Transformer Decoder is a powerful query-based semantic feature aggregator.

We also compute the similarity between the self-supervised prototypes P and the 21 semantic classes of PVOC (including "Background") and show the results after *softmax* in Table IV, each semantic class is represented by averaging the semantic prototypes manually extracted from 10 typical images, and very small similarity scores are ignored. Each prototype's largest similarity score is marked in bold, and the second largest score is underlined. Interestingly, the five learnt prototypes cover all 21 semantic classes with each prototype preferring certain classes. "Prototype 1-3" cover a wide range of semantics, while

TABLE IV
SIMILARITY BETWEEN THE LEARNT PROTOTYPES AND 21 SEMANTIC CLASSES OF PVOC

Prototype	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	TV/ Monitor	Background
1	0.1	0.	0.05	0.	0.	0.	0.08	0.1	0.	0.	0.	0.08	0.1	0.	0.	0.08	0.06	**0.27**	0.	0.05	0.03
2	0.06	0.04	0.	0.04	0.07	0.08	0.08	0.05	0.03	0.	0.06	0.04	0.03	0.05	0.11	0.04	0.	0.	0.06	**0.14**	0.02
3	0.	0.13	0.	0.10	0.	**0.35**	0.09	0.	0.	0.08	0.	0.	0.17	0.08	0.	0.	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.23	0.	**0.63**	0.	0.	0.14
5	0.	0.	0.	0.	0.19	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.15	0.	0.	0.	0.	**0.66**

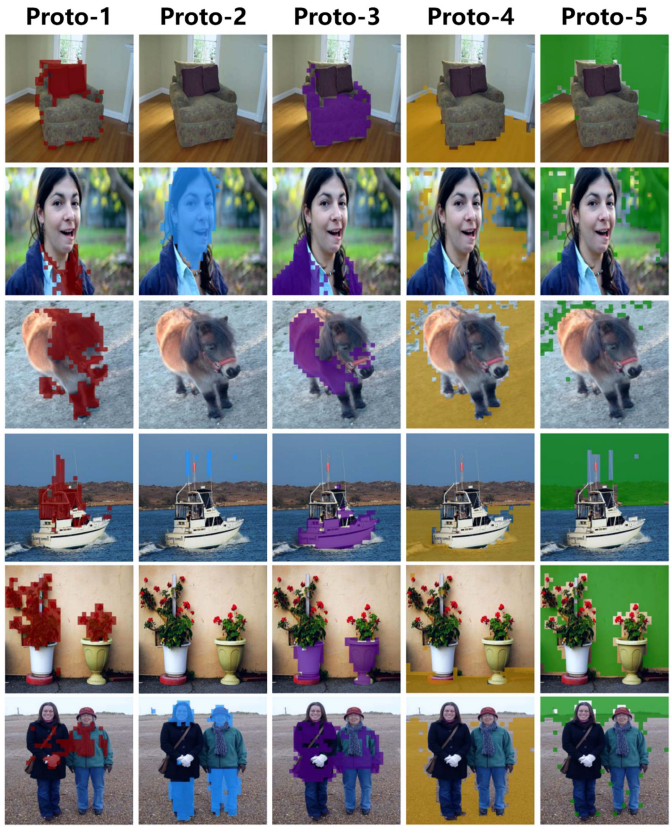

Fig. 8. Prototypical segmentation results of different types of object. Segmentation masks are marked with corresponding colors for different prototypes.

"Prototype 4" and "Prototype 5" are more similar to background or background related semantics (e.g. sofa, potted plant).

t-SNE visualization: We visualize the patch embeddings and the extracted prototypes, along with the segmentation regions by prototypes in Fig. 7. DINO pre-trained ViT already provides semantic-aware dense representations, from which AlignSeg successfully extracts the most semantically representative prototypes. And importantly, though background accounts for most of the patches, AlignSeg manages to focus on meaningful foreground patches by the *attention map constraint*. It is worth mentioning that number of semantic classes discovered by prototypes are more than the number of ground-truth classes for object-centric images. Taking the second image in Fig. 7 as an example, AlignSeg extracts four semantic prototypes (i.e. wall, ground, plant and pot), compared to only one semantic class (i.e. potted plant) in ground-truth annotation.

Difference from K-means clustering: Although K-means clustering can be applied directly to patch embeddings per image to obtain segmentations, the result is far from acceptable. Theoretically, K-means clustering forces image patches to be split into K clusters, without discriminating foreground and background patches, while AlignSeg adaptively extracts semantic prototypes with preference on foreground content and the number of segmentation masks are adaptive to the diversity of the image content as well. We visualize examplary segmentation results of AlignSeg and K-means clustering in Fig. 9, where the performance differences are significant. It further proves that the design of Transformer Decoder with learnable prototypes is superior semantic feature aggregator than traditional clustering.

Fig. 9. Comparison of segmentation results of AlignSeg and K-means clustering.

VI. CONCLUSION

In this paper, we propose a novel self-supervised semantic segmentation training and inferring paradigm where inferring is performed in a real-time and end-to-end manner. We first probe the dense representations by image-level self-supervised ViTs and reveal the semantic inconsistency between patches and the poor semantic quality in non-salient regions, then we propose *prototype-image alignment* and *global-local alignment* with *attention map constraint* to train a tailored Transformer Decoder with learnable prototypes and utilize adaptive prototypes for inference per image by a simple similarity *argmax* function. AlignSeg achieves state-of-the-art segmentation performance on PVOC and COCO-Stuff-27 under fully unsupervised semantic segmentation settings. AlignSeg can also be easily integrated with various self-supervised dense representation learning methods, and we prove the adaptability and effectiveness of the

proposed Transformer Decoder framework by probing the prototypes.

VII. Limitations

In this work, we break down self-supervised semantic segmentation into self-supervised dense representation learning and self-supervised segmentation head training, and we primarily focus on the latter based on fixed self-supervised image encoder. So the performance of the proposed AlignSeg heavily relies on the quality of the image encoder and its upper bound is also limited by the encoder. Moreover, when extending AlignSeg to different self-supervised methods, we find AlignSeg is very sensitive to the pre-trained encoders and requires delicate hyper-parameter tuning to realize a stable and effective training. Additionally, the source domain for pre-training the image encoder and the target domain for training AlignSeg should be homogeneous, otherwise AlignSeg would fail. Another limitation of AlignSeg is that once the model has been trained, the number of prototypes is fixed and AlignSeg cannot adapt to the actual number of semantic classes in an image when inferring. What is more, AlignSeg does not use any technical tricks for segmentation inference, e.g. CRF [45] or multi-scale strategy, thus the segmentation results may seem coarse-grained with the low resolution after patchification.

References

[1] X. Zou et al., "Generalized decoding for pixel, image, and language," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 15116–15127.

[2] J. Hwang et al., "SegSort: Segmentation by discriminative sorting of segments," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 7333–7343.

[3] X. Zhang and M. Maire, "Self-supervised visual representation learning from hierarchical grouping," in *Proc. Adv. Neural Inf. Process. Syst. 33: Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 16579–16590.

[4] W. V. Gansbeke, S. Vandenhende, S. Georgoulis, and L. V. Gool, "Unsupervised semantic segmentation by contrasting object mask proposals," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 10032–10042.

[5] R. R. Selvaraju, K. Desai, J. Johnson, and N. Naik, "Casting your model: Learning to localize improves self-supervised representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11058–11067.

[6] F. Wei, Y.Z. GaoH. WuHu, and S. Lin, "Aligning pretraining for detection via object-level contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst. 34: Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 22682–22694.

[7] J. X. XieZ. Zhan Liu, Y. S. Ong, and C. C. Loy, "Unsupervised object-level representation learning from scene images," in *Proc. Adv. Neural Inf. Process. Syst. 34: Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 28864–28876.

[8] O. J. Hénaff et al., "Object discovery and representation networks," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, 2022, pp. 123–143.

[9] J. H. Cho, U. Mall, K. Bala, and B. Hariharan, "Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Computer Vision Foundation, 2021, pp. 16794–16804.

[10] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 9630–9640.

[11] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman, "Unsupervised semantic segmentation by distilling feature correspondences," in *Proc. 10th Int. Conf. Learn. Representations*, 2022, pp. 1–26.

[12] K. Li et al., "ACSeg: Adaptive conceptualization for unsupervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 7162–7172.

[13] A. Ziegler and Y. M. Asano, "Self-supervised learning of object parts for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022. pp. 14482–14491.

[14] X. Wen, B. Zhao, A. Zheng, X. Zhang, and X. Qi, "Self-supervised visual representation learning with semantic grouping," in *Proc. Adv. Neural Inf. Process. Syst. 35: Annu. Conf. Neural Inf. Process. Syst.*, 2022, pp. 16423–16438.

[15] H. S. Seong, W. Moon, S. B. Lee, and J. Heo, "Leveraging hidden positives for unsupervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 19540–19549.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Springer, 2020, pp. 213–229.

[17] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst. 34: Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 17864–17875.

[18] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[19] M. Heo, S. Hwang, S. W. Oh, J. Lee, and S. J. Kim, "VITA: Video instance segmentation via object token association," in *Proc. Adv. Neural Inf. Process. Syst. 35: Annu. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2022.

[20] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[21] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Springer, 2014, pp. 740–755.

[22] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv: 1807.03748*. [Online]. Available: http://arxiv.org/abs/1807.03748

[23] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 9620–9629.

[24] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 15979–15988.

[25] Z. Xie et al., "Simmim: A simple framework for masked image modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 9643–9653.

[26] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Computer Vision Foundation, 2021, pp. 3024–3033.

[27] X. Li et al., "Dense semantic contrast for self-supervised visual representation learning," in *Proc. ACM Multimedia Conf.*, China, 2021, pp. 1368–1376.

[28] P. O. Pinheiro, A. Almahairi, R. Y. Benmalek, F. Golemo, and A. C. Courville, "Unsupervised learning of dense visual representations," in *Proc. Adv. Neural Inf. Process. Syst. 33: Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 4489–4500.

[29] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Computer Vision Foundation, 2021, pp. 16684–16693.

[30] Z. Yin et al., "Transfgu: A top-down approach to fine-grained unsupervised semantic segmentation," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, 2022, pp. 73–89.

[31] L. Melas-Kyriazi, C. Rupprecht, I. Laina, and A. Vedaldi, "Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 8354–8365.

[32] X. Wang et al., "FreeSOLO: Learning to segment objects without annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 14156–14166.

[33] O. Siméoni et al., "Localizing objects with self-supervised transformers and no labels," in *Proc. 32nd Brit. Mach. Vis. Conf.*, BMVA Press, 2021, Art. no. 310.

[34] W. V. Gansbeke, S. Vandenhende, and L. V. Gool, "Discovering object masks with transformers for unsupervised semantic segmentation," 2022, *arXiv:2206.06363*. [Online]. Available: https://doi.org/10.48550/arXiv.2206.06363

[35] S. Gao, Z. Li, M. Yang, M. Cheng, J. Han, and P. H. S. Torr, "Large-scale unsupervised semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7457–7476, Jun. 2023.

[36] J. Xu et al., "GroupViT: Semantic segmentation emerges from text supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 18113–18123.
[37] H. Luo, J. Bao, Y. Wu, X. He, and T. Li, "SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation," in *Proc. Int. Conf. Mach. Learn.*, Honolulu, Hawaii, USA, 2023, pp. 23033–23044.
[38] Y. Liu, N. Liu, X. Yao, and J. Han, "Intermediate prototype mining transformer for few-shot semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst. 35: Annu. Conf. Neural Inf. Process. Syst.*, 2022, pp. 38020–38031.
[39] H. Ma, X. Lin, and Y. Yu, "I2F: A unified image-to-feature approach for domain adaptive semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 3, pp. 1695–1710, Mar. 2024.
[40] C. Liang, W. Wang, J. Miao, and Y. Yang, "GMMSeg: Gaussian mixture based generative semantic segmentation models," in *Proc. Adv. Neural Inf. Process. Syst. 35: Annu. Conf. Neural Inf. Process. Syst.*, 2022, pp. 31360–31375.
[41] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 1280–1289.
[42] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, no. 1/2, pp. 83–97, 1955.
[43] X. Ji, A. Vedaldi, and J. F. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 9864–9873.
[44] M. Seitzer et al., "Bridging the gap to real-world object-centric learning," in *Proc. 11th Int. Conf. Learn. Representations*, Kigali, Rwanda, 2023, pp. 1–43.
[45] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst. 24: 25th Annu. Conf. Neural Inf. Process. Syst.*, Granada, Spain, 2011, pp. 109–117.
[46] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv: 1706.05587*.
[47] A. Kirillov et al., "Segment anything," 2023, *arXiv:2304.02643*.
[48] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," 2023, *arXiv:2304.07193*.

Yue Liu received the master's degree in computer science from the City University of Hong Kong, in 2014. He is currently working toward the PhD degree with Guangzhou University, supervised by Prof. Gang Fang. His research interests include computer vision, and self-supervised learning.

Jun Zeng received the master's degree in traffic information engineering and control from East China Jiaotong University, in 2012. He is now a lecturer with the Jiangxi College of Applied Technology. His research interests include artificial intelligence, deep learning, and cloud computing.

Xingzhen Tao received the master's degree in computer software and theory from the Guilin University of Electronic Technology, in 2014. She is now an associate professor with the Jiangxi College of Applied Technology. Her research interests include computer vision, and deep learning.

Gang Fang received the PhD degree from the Huazhong University of Science and Technology, in 2006. He worked as a visiting scholar with Virginia Polytechnic Institute and State University from 2013 to 2014. He is now a professor with Guangzhou University. His research interests include artificial intelligence, and biological computing.