AllSpark: Reborn Labeled Features from Unlabeled in Transformer for Semi-Supervised Semantic Segmentation

Haonan Wang,* Qixiang Zhang,* Yi Li, Xiaomeng Li✉

The Hong Kong University of Science and Technology

{hwanggr, qzhangcq, ylini}@connect.ust.hk, eexmli@ust.hk

Abstract

*Semi-supervised semantic segmentation (SSSS) has been proposed to alleviate the burden of time-consuming pixel-level manual labeling, which leverages limited labeled data along with larger amounts of unlabeled data. Current state-of-the-art methods train the labeled data with ground truths and unlabeled data with pseudo labels. However, the two training flows are separate, which allows labeled data to dominate the training process, resulting in low-quality pseudo labels and, consequently, sub-optimal results. To alleviate this issue, we present **AllSpark**[1], which reborns the labeled features from unlabeled ones with the channel-wise cross-attention mechanism. We further introduce a Semantic Memory along with a Channel Semantic Grouping strategy to ensure that unlabeled features adequately represent labeled features. The **AllSpark** shed new light on the architecture level designs of SSSS rather than framework level, which avoids increasingly complicated training pipeline designs. It can also be regarded as a flexible bottleneck module that can be seamlessly integrated into a general transformer-based segmentation model. The proposed **AllSpark** outperforms existing methods across all evaluation protocols on Pascal, Cityscapes and COCO benchmarks without bells-and-whistles. Code and model weights are available at: https://github.com/xmed-lab/AllSpark.*

1. Introduction

Semantic segmentation is a dense prediction task that aims to enable deep learning models to have class awareness [5, 30, 33, 48]. However, the success of such approaches heavily relies on labor-intensive manual labeling. Semi-Supervised Semantic Segmentation (SSSS) methods [6, 36,

---

* Equal contribution.
✉ Corresponding author.
[1] The *AllSpark* is a powerful Cybertronian artifact in the film of *Transformers: Revenge of the Fallen*, which can be used to reborn the Transformers. It aligns well with our core idea.
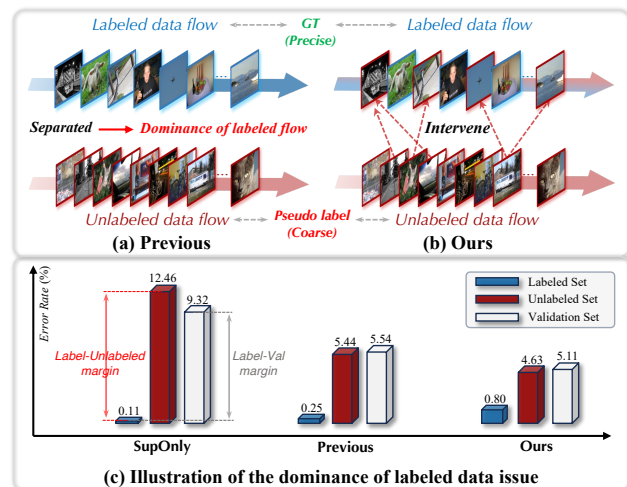

Figure 1. (a)(b) Comparison between the training data flows of previous methods and ours. Previous methods separate the labeled and unlabeled data training flows thus leading to dominance of the labeled data. (c) Dominance of labeled data issue of the previous method (*e.g.*, UniMatch [50]). The red margin denotes how the labeled data overwhelm the unlabeled; Larger gray margin indicates the model may over-fit to labeled data.

40, 41, 45, 50, 59], which are built upon the principles of semi-supervised learning [1, 22, 34, 54], are well-studied to train models using a limited amount of labeled data in conjunction with a larger amount of unlabeled data.

Most state-of-the-art SSSS methods [23, 24, 26, 31, 50] are based on the pseudo labeling [22, 34] scheme, where the unlabeled data is assigned pseudo labels based on model predictions. The model is then iteratively trained using these pseudo labeled data as if they were labeled examples. Based on the training flows of the labeled and unlabeled data, the current state-of-the-arts can be mainly divided into two categories: (1) *successive*, the most representative one is the teacher-student-based framework [17, 20, 36, 44, 45, 56], where a student model is first trained with labeled data, then a teacher model updated by the exponential moving av-
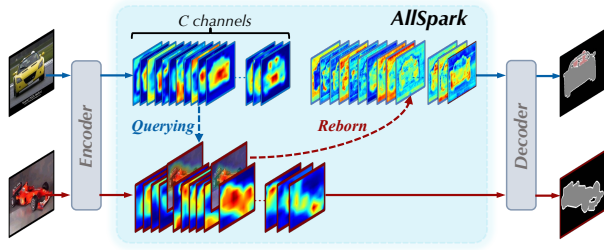
Figure 2. Illustration of the core idea of **AllSpark**, which leverages the unlabeled features to reborn the labeled ones. The regenerated labeled features exhibit a high level of precision, yet they also possess diversity compared to the original features.

eraging (EMA) of the student is used to generate the pseudo labels; (2) *parallel*, represented by CPS (Cross Pseudo Supervision) [6, 23, 46, 50], where the labeled data are concatenated with the unlabeled data within a mini-batch, and the model optimize the two flows simultaneously through consistency regularization.

One key characteristic of these state-of-the-art methods is the separation of training flows for labeled and unlabeled data, as shown in Figure 1(a). The separation of training flows allows labeled data to dominate the training process since they have labels; see Figure 1(c). However, these approaches have a common significant drawback: the model used to generate pseudo labels for unlabeled images is heavily influenced by a limited set of labeled samples, resulting in low-quality pseudo labels and, consequently, sub-optimal results. Unfortunately, previous research has overlooked this crucial observation.

To this end, we present a novel approach: rather than training directly on the labeled samples, we leverage the unlabeled ones to generate labeled features (as depicted in Figure 1(b)). The underlying rationale is that the labeled data is far less than the unlabeled data, by reborning labeled features from unlabeled data, we introduce diversity into the labeled flow, creating a more challenging learning environment. Instead of relying solely on self-reconstruction, the labeled branch learns to extract valuable channels from the unlabeled features. The generic semantic information within channels provides feasibility to this idea, for example, as illustrated in Figure 3, the labeled sample of a "boat" and the unlabeled sample of "airplanes" exhibit a similar metal material texture.

The subsequent step is to determine an effective method to reborn the labeled data. Fortunately, the Cross-Attention mechanism [39], initially utilized in the transformer decoder, provides a suitable approach as it is designed to reconstruct a target sequence using the source sequence. Therefore, we develop our method based on the concept of cross-attention, adapting it to operate in a channel-wise manner [9, 42] in order to extract channel-wise semantic

information. This method, named **AllSpark**, utilizes the labeled data features as queries and the unlabeled data features as keys and values, as depicted in Figure 2. Specifically, **AllSpark** calculates a similarity matrix between each channel of the labeled features and the unlabeled features. The unlabeled channels with the highest similarities are then emphasized to reconstruct the labeled features. However, a critical challenge in implementing this idea is the limited unlabeled features within a mini-batch, which may not be sufficient to reconstruct the correct labeled features. To address this, we introduce a Semantic Memory (S-Mem) that gathers channels from previous unlabeled features. S-Mem are updated iteratively with a proposed channel-wise semantic grouping strategy which first groups together channels that represent the same semantics and then enqueue them into according class slot of S-Mem.

Furthermore, naively plugging **AllSpark** into a CNN-based backbone is not applicable, due to the smaller reception field of the convolution operation, as shown in Figure 8, compared with that of the Vision Transformers (ViTs) [10], which have made remarkable progress in semantic segmentation [7, 18, 19, 35, 43, 48] with the well-known property of capturing long-range dependencies. Thus, we focus on building a pure-transformer-based SSSS framework.

Our contributions can be summarized in three folds:

- We notice the separate training approach in current SSSS frameworks will cause the dominance of labeled data and thus lead to sub-optimal solutions.
- We propose **AllSpark** module to address the labeled data dominance issue and build a SOTA pure-transformer-based SSSS framework, which can further be benefit for efficiently training the foundation models, *e.g.*, SAM [21], with less labeled data.
- Extensive experiments have been conducted to validate the effectiveness of our proposed **AllSpark**. The results of these experiments have demonstrated solid performance gains across three widely recognized benchmarks: PASCAL VOC 2012 [11], Cityscapes [8], and COCO [27].

2. Related Work

2.1. Semantic Segmentation

Semantic segmentation is a dense prediction tasks, and has been significantly advanced in many real-world applications [4, 5, 30, 33, 55]. Most recently, Vision Transformers (ViTs) [2, 3, 10, 13, 29, 43, 47, 51, 53] have remarkably reformed this field. Concretely, SETR [57] adopts ViT as a backbone to extract features, achieving impressive performance. PVT [43] is the first work to introduce a pyramid structure in Transformer. SegFormer [48], with hierarchical MiT encoder and a lightweight All-MLP decoder, serves as a solid baseline for semantic segmentation. These ViT-based segmentation approaches demonstrate the potential of
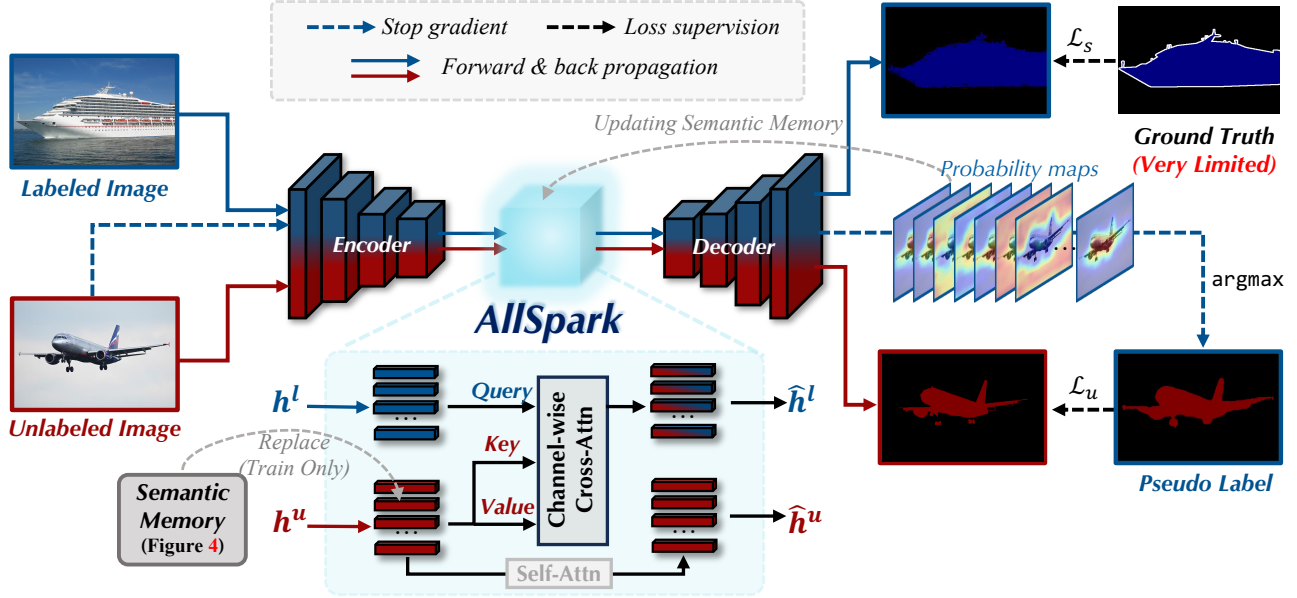
Figure 3. Illustration of the proposed *AllSpark*, which can be regarded as a flexible bottleneck plugged in the middle of a general segmentation model. In the training stage, the unlabeled features are replaced by the Semantic Memory (§ 3.3 & Figure 4 bottom). Moreover, the probability maps is used for Channel Semantic Grouping strategy (§ 3.4 & Figure 4 top). In the inference stage, the cross-attention is degraded to self-attention with the inputs as the hidden features of the test images.

a pure Transformer framework in dense prediction tasks. However, when it comes to semi-supervised setting, it is hard for such a framework to achieve the same advances. The reason is that these Transformer-based methods have weaker inductive bias compared with CNN-based methods and heavily rely on large amount of training data [10]. Thus, appropriate designs are desirable to train these methods in low-data regime.

2.2. Semi-supervised Semantic Segmentation

Conventional semi-supervised semantic segmentation (SSSS) methods focus on how to better leverage the unlabeled data with framework level designs, and use basic **CNNs-based** segmentation model as backbone. Two most remarkable basic frameworks are the teacher-student [20, 36, 44, 45, 56] and CPS (Cross Pseudo Supervision) [6, 46, 50]. Specifically, in the teacher-student-based framework, a student model is first trained with labeled data, then a teacher model updated by the exponential moving averaging (EMA) of the student is used to generate the pseudo labels. In the CPS-based framework, the labeled data are concatenated with the unlabeled data within a mini-batch, and the model optimize the two flows simultaneously with consistency regularization.

Recently, several works have explored **Transformer-based** models to extend the border of SSSS. Specifically, SemiCVT [17] presents an inter-model class-wise consistency to complement the class-level statistics of CNNs

and Transformer in a cross-teaching manner. Other methods [23, 25] utilize ViTs as a diverse model to CNNs in CPS-based framework [6]. These methods can be summarized as CNN-Transformer-based methods and the way they utilize transformer are naive, hence restrict the full potential of ViT. Unlike these methods, we shed new light on a pure-transformer SSSS method with architecture level designs.

3. *AllSpark*: Reborn Labeled Features from Unlabeled Features

3.1. Baseline and Overview

General semi-supervised semantic segmentation dataset comprises limited labeled data and plenty of unlabeled data. Assume that the entire dataset comprises of N_L labeled samples $\{(x_i^l, y_i)\}_{i=1}^{N_L}$ and N_U unlabeled samples $\{x_i^u\}_{i=1}^{N_U}$, where $x_i \in \mathbb{R}^{3 \times H \times W}$ is the input volume and $y_i \in \mathbb{R}^{K \times H \times W}$ is the ground-truth annotation with K classes. Pseudo-labeling-base methods train the model with ground truth supervision first, then generate pseudo labels $\hat{y}$ with the pretrained model as supervision to the unlabeled data. The most basic objective function is formulated as:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u = \frac{1}{N_L} \sum_{i=0}^{N_L} \mathcal{L}_{CE}(p_i^l, y_i) + \frac{1}{N_U} \sum_{j=0}^{N_U} \mathcal{L}_{CE}(p_j^u, \hat{y}_j)$$
(1)

where $\mathcal{L}_{CE}$ is the cross-entropy loss.

Increasing methods have been proposed to extend this

formulation, especially for the unsupervised loss, making this formulation and the training pipeline complicated. In this work, we focus on introducing *architecture-level* modifications to enhance the performance of pseudo-labeling in semi-supervised semantic segmentation. We start with the basic formulation of pseudo-labeling as our baseline and then propose the ***AllSpark*** module, which is integrated between the encoder and decoder of the ViT-based segmentation model. The ***AllSpark*** module, as illustrated in Fig. 3 and Algorithm 1, consists a Channel-wise Cross-Attention module (§3.2), a Semantic Memory to store unlabeled features for better reconstruction of the labeled features (§3.3), and a Channel-wise Semantic Grouping strategy to better update the Semantic Memory (§3.4).

3.2. Channel-wise Cross-attention: the Core of *AllSpark*

To alleviate the dominance of the biased labeled data brought by the separate training scheme, we propose ***AllSpark*** to directly intervene in the labeled data training flow with unlabeled data. Typically, different feature channels encode distinct semantic information. Compared to patch tokens, these channel-wise features encompass richer contextual information that is more general across various input images (refer to Figure 7). We leverage this contextual information from unlabeled data to reconstruct the features of labeled data as a robust regularization using the channel-wise cross-attention mechanism [9, 42]. Within this mechanism, labeled data features serve as queries, while unlabeled features serve as keys and values, as depicted in Figure 3. Specifically, we calculate the similarity between each channel of the labeled features and the unlabeled features, and the unlabeled channels with the highest similarities play a more significant role in reconstructing the labeled features. The underlying rationale is that even though the unlabeled features may originate from different classes compared to the labeled features, they still possess generic information with channel wise that can be shared, such as textures.

Given the hidden features $[h^l, h^u]$ of labeled and unlabeled data after the encoder stage, we split them then set h^l as *query* and h^u as *key* and *value* in multi-head manner:

$$q = h^l w_q, k = h^u w_k, v = h^u w_v \quad (2)$$

where $w_q, w_k, w_v \in \mathbb{R}^{C \times 2C}$ are transformation weights, $h^l, h^u \in \mathbb{R}^{C \times d}$, d is the sequence length (number of patches) and C is the channel dimension. The channel-wise attention is defined as:

$$\hat{h}^l = \mathbf{M} v^\top = \sigma[\psi(q^\top k)] v^\top w_{out} \quad (3)$$

where $\psi(\cdot)$ and $\sigma(\cdot)$ denote the instance normalization [37] and the softmax function, $w_{out} \in \mathbb{R}^{2C \times C}$. In contrast to the traditional self-attention mechanism, channel-wise at-
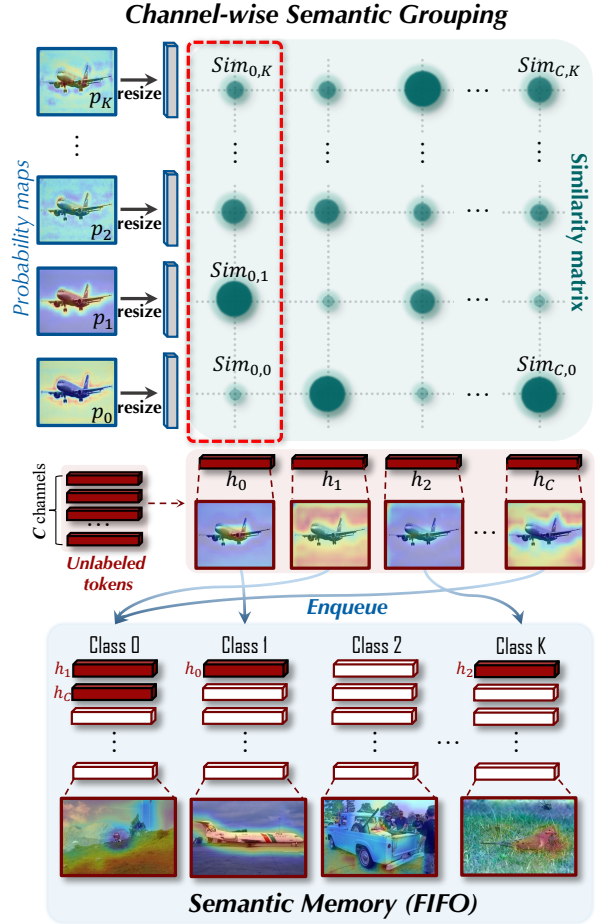

Figure 4. Illustration of the Class-wise Semantic Memory Bank (§3.3) and the Channel-wise Semantic Grouping (§3.4). $Sim_{i,j}$ denotes the similarity between i^{th} channel of the unlabeled hidden feature and the jth probability map. The dash lines give some visual examples of some channels. Take the column with red box as an example, h_0 has the largest similarity with p_1 ($Sim_{0,1}$), so it should be added to Class 1 slot of the Semantic Memory.

tention can capture long-range dependencies among channels. To further refine the hidden features of the unlabeled data, we also apply a channel-wise self-attention. The formulation to obtain the refined unlabeled feature $\hat{h}_i^u$ is similar to Eq. 2 & 3, with the only difference being that q is replaced with $q = h^u w_q$. The reconstructed $\hat{h}_i^l$ and $\hat{h}_i^u$ are subsequently fed into the decoder to generate the final predictions.

3.3. Semantic Memory for Enlarging the Feature Space of *AllSpark*

Directly utilizing the unlabeled features within a single mini-batch is not enough to effectively reborn the labeled features. To overcome this limitation, we need to expand

Algorithm 1 PyTorch-style pseudocode for *AllSpark* module.

```
# Take Pascal dataset as an example, where the
  crop size is 513 × 513
# channel num = 512, token num = 289, class num=21

# p_u ∈ ℝ^{289×21}:  probability token
p_u = flatten(decoder.forward(h_u).resize(17, 17))

# h_l ∈ ℝ^{289×512}:  labeled hidden feature map
# h_u ∈ ℝ^{289×512}:  unlabeled hidden feature map
def cross_attn((h_l,y), h_u, p_u):
    # channel2class similarity:  21 × 512
    sim_prob_chl = mm(p_u.T, h_u) # see eq (4)
    # channel pseudo label:  1 × 512
    chl_pseu = argmax(sim_prob_chl, dim=0)
    for cls in range(21):
        # group channels belonging to class cls
        index = nonzero(where(chl_pseu==cls, 1, 0))
        chls = h_u(:, index)
        N_u = len(index)
        # enqueue channels into semantic memory
        enqueue(S-Mem, chls, cls)
        dequeue(S-Mem, cls)
    # queries:  289 × 512
    q = l_q.forward(h_l)
    # Keys:  289 × 512
    k = l_k.forward(flatten(S-Mem(:,:,:N_u)))
    # values:  289 × 512
    v = l_v.forward(flatten(S-Mem(:,:,N_u:)))
    # channel similarity:  512 × 512
    m = mm(q.T, k)
    # reborn labeled feature:  289 × 512
    q_re = mm(m, v.T).T
    return q_re
```

the unlabeled feature space. To achieve this, we introduce a first-in-first-out queue, Class-wise Semantic Memory (S-Mem), to store a significant number of unlabeled features, as shown in Figure 4 bottom. This memory allows us to efficiently access a broader range of unlabeled features for the reconstruction process.

S-Mem has a shape of $\mathbb{R}^{K\times C\times d}$, where K represents the number of classes. For each class, S-Mem stores C channels, and each channel consists of d patches. During the training process, we utilize the semantic memory to replace the original *key* and *value* components of the unlabeled features (k^T and v^T in Eq. 3). In the next sub-section §3.4, we demonstrate how to update each class slot with accurate channels containing class-specific semantic information.

3.4. Channel-wise Semantic Grouping

Storing previous unlabeled features in a naive manner is not suitable for semantic segmentation tasks, as classes are often imbalanced. To ensure sufficient semantic information for each class, it is necessary to establish a class-balanced semantic memory. Thus, we introduce a Channel-wise Semantic Grouping strategy to determine the semantic representation of each channel in the unlabeled feature, and then group them and add into the corresponding class slot of S-Mem, as illustrated in Figure 4. Specifically, we calculate the similarity between the unlabeled feature $h^u \in \mathbb{R}^{C\times d}$ and the probability token $\hat{p} \in \mathbb{R}^{K\times d}$. The probability to-

ken is resized and reshaped vector obtained from the probability map $p \in \mathbb{R}^{K\times H\times W}$, which is the soft prediction of the segmentation network and contains the overall semantic information of h^u. Once the semantic representation is determined for each channel, we can store them in the memory according to their respective semantic categories. The similarity matrix $Sim \in \mathbb{R}^{K\times C}$ is defined as: $Sim_{i,j} = \psi(\hat{p}_j^\top h_i^u)$. In this way, by comparing the semantic information of $i^{th}, i \in [0, C]$ channel and the overall semantic context of the image, we could determine the most likely semantic category that the ith channel represents. Then, we group together the channels that represent the same semantic category and enqueue them into the appropriate class slot within the semantic memory.

In the inference stage, although we can still utilize the cross-attention mechanism with S-Mem, it often comes with a computational burden and provides only marginal improvements in performance. To enhance efficiency, we opt to remove S-Mem and CSG in the inference stage. As a result, the two inputs of cross-attention become identical, which can be simplified to self-attention.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our *AllSpark* on three widely-used datasets: **PASCAL VOC 2012** [11] is a dataset consisting of around 4,000 samples. This dataset is partitioned into three subsets: train, validation, and test. The train set comprises 1,464 images, the validation set contains 1,449 images, and the test set consists of 1,456 images. This dataset includes pixel-level annotations for a total of 21 categories, including the background. In accordance with the conventions established in [6, 45], an additional set of 9,118 coarsely labeled images from the SBD dataset [14] is incorporated as a complement to the training data, which is referred to as the *augmented* set. The **Cityscapes** dataset [8], is specifically designed for the purpose of urban scene understanding. It comprises a collection of 5,000 images that have been meticulously annotated with fine-grained pixel-level details. These images are divided into three sets: train, validation, and test. The train set contains 2,975 images, the validation set consists of 500 images, and the test set includes 1,524 images. In the Cityscapes dataset, there are 19 semantic categories that are organized into 6 superclasses. **COCO** [27] stands out for its extensive annotations and wide variety of object categories. The dataset includes 118,000 training images and 5,000 validation images. It covers 80 object categories and encompasses both indoor and outdoor scenes.

Implementation Details. For the three datasets, we all utilized SGD as the optimizer, and the poly scheduling to adjust the learning rate as $lr = lr_{init} \cdot (1 - \frac{i}{I})^{0.9}$, where lr_{init}

PASCAL *original*	1/16 (92)	1/8 (183)	1/4 (366)	1/2 (732)	Full(1464)
SupBaseline-CNN	45.1	55.3	64.8	69.7	73.5
SupBaseline-ViT	50.65	63.62	70.76	75.44	77.01
CutMixSeg [12] [BMVC'20]	55.58	63.20	68.36	69.84	76.54
PseudoSeg [59] [ICLR'21]	57.60	65.50	69.14	72.41	
CPS [6] [CVPR'21]	64.07	67.42	71.71	75.88	-
PC2Seg [58] [ICCV'21]	57.00	66.28	69.78	73.05	74.15
PS-MT [28] [CVPR'22]	65.80	69.58	76.57	78.42	80.01
ST++ [49] [CVPR'22]	65.20	71.00	74.60	77.30	79.10
U²PL [45] [CVPR'22]	67.98	69.15	73.66	76.16	79.49
GTA-Seg [20] [NeurIPS'22]	70.02	73.16	75.57	78.37	80.47
SemiCVT [17] [CVPR'23]	68.56	71.26	74.99	78.54	80.32
FPL [32] [CVPR'23]	69.30	71.72	75.73	78.95	-
CCVC [46] [CVPR'23]	70.2	74.4	77.4	79.1	80.5
AugSeg [56] [CVPR'23]	71.09	75.45	78.8	80.33	81.36
DGCL [44] [CVPR'23]	70.47	77.14	78.73	79.23	81.55
UniMatch [50] [CVPR'23]	75.2	77.19	78.8	79.9	-
ESL [31] [ICCV'23]	70.97	74.06	78.14	79.53	81.77
LogicDiag [26] [ICCV'23]	73.25	76.66	77.93	79.39	-
AllSpark (Ours)	**76.07**	**78.41**	**79.77**	**80.75**	**82.12**

Table 1. Comparison with SOTAs on **Pascal** *original* dataset.

Cityscapes	1/16 (186)	1/8 (372)	1/4 (744)	1/2 (1488)
SupBaseline-CNN	66.3	72.8	75.0	78.0
SupBaseline-ViT	65.69	70.52	74.01	78.14
CPS [6] [CVPR'21]	69.78	74.31	74.58	76.81
AEL [16] [NeurIPS'21]	74.45	75.55	77.48	79.01
ST++ [49] [CVPR'22]	67.64	73.43	74.64	77.78
U²PL [45] [CVPR'22]	74.90	76.48	78.51	79.12
GTA-Seg [20] [NeurIPS'22]	69.38	72.02	76.08	-
SemiCVT [17] [CVPR'23]	72.19	75.41	77.17	79.55
FPL [32] [CVPR'23]	75.74	78.47	79.19	-
CCVC [46] [CVPR'23]	74.9	76.4	77.3	-
AugSeg [56] [CVPR'23]	75.22	77.82	79.56	80.43
DGCL [44] [CVPR'23]	73.18	77.29	78.48	80.71
UniMatch [50] [CVPR'23]	76.6	77.9	79.2	79.5
ESL [31] [ICCV'23]	75.12	77.15	78.93	80.46
LogicDiag [26] [ICCV'23]	76.83	78.90	80.21	81.25
CFCG [24] [ICCV'23]	77.28	79.09	80.07	80.59
AllSpark (Ours)	**78.33**	**79.24**	**80.56**	**81.39**

Table 2. Comparison with SOTAs on **Cityscapes** dataset.

PASCAL *augmented*	1/16 (662)	1/8 (1323)	1/4 (2646)	1/2 (5291)
SupBaseline-CNN	67.5	71.1	74.2	-
SupBaseline-ViT	72.01	73.20	76.62	77.61
CutMixSeg [12] [BMVC'20]	72.56	72.69	74.25	75.89
PseudoSeg [59] [ICLR'21]	-	-	-	-
AEL [16] [NeurIPS'21]	77.20	77.57	78.06	80.29
CPS [6] [CVPR'21]	72.18	75.83	77.55	78.64
PC2Seg [58] [ICCV'21]	-	-	-	-
PS-MT [28] [CVPR'22]	75.50	78.20	78.72	79.76
ST++ [49] [CVPR'22]	74.70	77.90	77.90	-
FPL [32] [CVPR'23]	74.98	77.75	78.30	-
CCVC [46] [CVPR'23]	77.2	78.4	79.0	-
AugSeg [56] [CVPR'23]	77.01	78.20	78.82	-
DGCL [44] [CVPR'23]	76.61	78.37	79.31	80.96
UniMatch [50] [CVPR'23]	78.1	78.4	79.2	-
ESL [31] [ICCV'23]	76.36	78.57	79.02	79.98
CFCG [24] [ICCV'23]	76.82	79.10	79.96	80.18
DLG [23] [ICCV'23]	77.75	79.31	79.14	79.54
AllSpark (Ours)	**78.32**	**79.98**	**80.42**	**81.14**
U²PL† [45] [CVPR'22]	77.21	79.01	79.30	80.50
GTA-Seg† [20] [NeurIPS'22]	77.82	80.47	80.57	81.01
SemiCVT† [17] [CVPR'23]	78.20	79.95	80.20	80.92
UniMatch† [50] [CVPR'23]	80.94	81.92	80.41	-
LogicDiag† [26] [ICCV'23]	79.65	80.24	80.62	81.00
AllSpark† (Ours)	**81.56**	**82.04**	**80.92**	**81.13**

Table 3. Comparison with SOTAs on the **Pascal** *augmented* dataset. † means using the same split as U²PL [45].

COCO	1/512 (232)	1/256 (463)	1/128 (925)	1/64 (1849)
SupBaseline-CNN	22.94	27.96	33.60	37.80
SupBaseline-ViT	19.66	26.72	35.90	40.89
PseudoSeg [59] [ICLR'21]	29.78	37.11	39.11	41.75
PC2Seg [58] [ICCV'21]	29.94	37.53	40.12	43.67
MKD [52] [ACM MM'23]	30.24	38.04	42.32	45.50
UniMatch [50] [CVPR'23]	31.86	38.88	44.35	48.17
LogicDiag [26] [ICCV'23]	33.07	40.28	45.35	48.83
AllSpark (Ours)	**34.10**	**41.65**	**45.48**	**49.56**

Table 4. Comparison with SOTAs on different partitions of the **COCO** dataset.

represents the initial learning rate, i is the current iteration, and I is the max number of iterations. For the *PASCAL VOC 2012 dataset*, we set the initial learning rate as 0.001. Please note that the learning rate for the AllSpark and the decoder is 5 times of the backbone network. We train the model for 80 epochs. For the *Cityscapes dataset*, the initial learning rate is set as 0.005, and we train the model for 240 epochs. It is worth noting that for the Cityscapes dataset, we employ two approaches used in previous works [6, 16, 28, 45, 50], i.e., online hard example mining (OHEM) and sliding window evaluation. For the *COCO dataset*, we set the initial learning rate of 0.001, and train the model for 10 epochs.

Baselines. We report two supervised baselines among different ratios of labeled data: a CNN-based model, RestNet-101 [15] with DeepLabV3+ [5], referred to as **SupBaseline-CNN**, and a ViT-based model, SegFormer-B5 [48], referred to as **SupBaseline-ViT**. Additionally, we employ a simple pseudo-labeling scheme outlined in section §3.1 with

SegFormer-B5 as our semi-supervised baseline, denoted as **SemiBaseline**.

4.2. Comparison with Existing Methods

PASCAL VOC 2012 original. Table 1 presents a summary of the quantitative comparisons across different label amounts. From this analysis, we make an important observation: our method consistently outperforms the previous state-of-the-art approaches in all scenarios. This highlights the effectiveness of *AllSpark* and establishes it as the new state-of-the-art.

PASCAL VOC 2012 augmented. Table 3 showcases the comparison results using the augmented training set. Once again, our method establishes new state-of-the-art performance across all partition protocols. It achieves an average gain of 0.47. Furthermore, we conducted a comparison using the same setting as U²PL [45], where all labeled data is of high quality. In this scenario, *AllSpark* demonstrates an
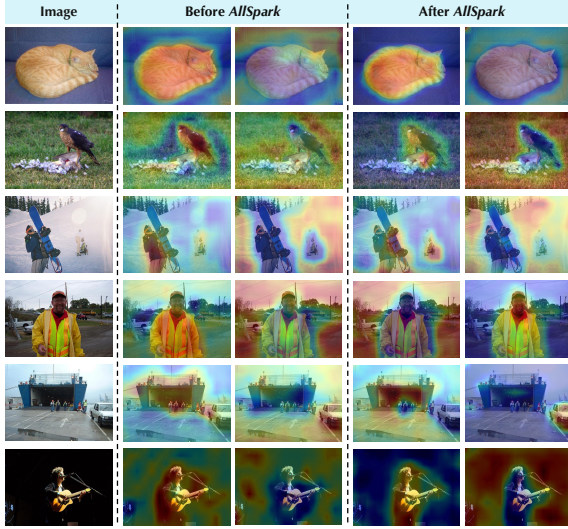
Figure 5. Visualization of labeled feature channels before and after the ***AllSpark*** with the same indexes. The features before *AllSpark* focus on more similar regions, while those after *AllSpark* focus on different objects or context.
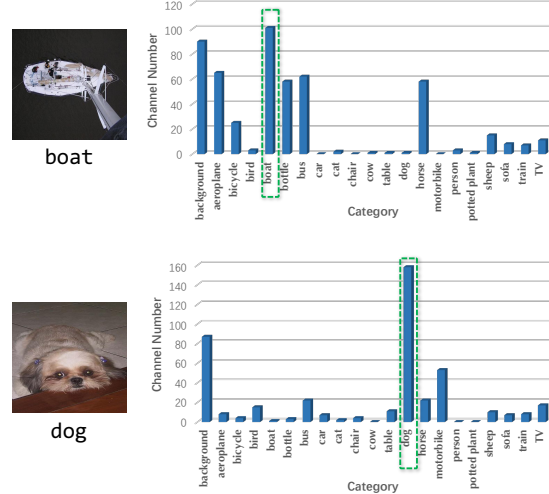

Figure 6. Ablation on the effectiveness of the similarity-guided channel grouping strategy. The y-axis of the bar chart denotes the number of channels to be selected and added to the corresponding slot of the Semantic Memory.

even greater performance gain.

Cityscapes. Table 2 quantitatively compares our method against the competitors on Cityscapes val. In spite of the presence of complex street scenes, our method still delivers a solid overtaking trend across different partitions.

COCO. In Table 4, we showcase the model performance on the COCO validation set. Notably, leveraging the extensive semantic hierarchy present in COCO, ***AllSpark*** achieves significant performance gains over the leading method in all partitions. These experimental results further validate the effectiveness of ***AllSpark***.

| SemiBaseline | ***AllSpark*** | | | 1/8 | 1/2 |
	CCA	S-Mem	CSG	(183)	(732)
✓				68.71	74.40
✓	✓			76.77	78.93
✓	✓	✓		77.62	79.71
✓	✓	✓	✓	**78.41**	**80.75**

Table 5. Effectiveness of the proposed components on **PASCAL** dataset. We leverage a basic pseudo-labeling scheme (§3.1) with the SegFormer [48] as our baseline, ***AllSpark*** consists of 3 hierarchy components: CCA (Channel-wise Cross-Attention), S-Mem (Semantic Memory), and CSG (Channel Semantic Grouping).

4.3. Ablation Studies

Effectiveness of the *AllSpark* Module. We analyze the effectiveness of different components in ***AllSpark***, *i.e.*, the channel-wise cross attention mechanism (CCA), semantic memory (S-Mem) and channel-wise similarity-guided grouping strategy (CSG) as in Table 5. The three com-

ponent are hierarchy, *i.e.*, S-Mem is built upon CCA, the core of ***AllSpark***, and CSG strategy is built for the update of S-Mem. According to the results, the core of ***AllSpark***, channel-wise cross-attention mechanism (the second row), helps our method outperform the previous methods, pushing the performance about 8.06%/4.53% higher than the baseline (the first row). Further, the semantic memory brings about 0.85%/0.78% performance improvements. With all of these components, our method outperforms the baseline by over 9.70%/5.95% in mIoU. Furthermore, we also provide visual verification of ***AllSpark*** module, as shown in Figure 5. The reconstructed labeled features after ***AllSpark*** are quite diverse compared with the original features, which is useful for alleviating the dominance of labeled data. Moreover, the features tend to be closer to the target due to the larger feature space compared with a self-attention with the features of itself.

The necessity of the Semantic Memory for enlarging unlabeled feature space. As shown in Figure 7, we visualize part of the Semantic Memory. Overall, the Semantic Memory stores quite accurate features related to the class slot and indeed enlarges the unlabeled feature space not only in the same classes with the current labeled data, but also the features from other classes as negative samples when calculating the similarity matrix.

Ablation on the effectiveness of the Channel-wise Semantic Grouping strategy. As shown in Table 5, compared with blindly adding all channels to the semantic memory, using semantic grouping strategy obtains 0.79%/0.64% performance gain. Figure 6 also verifies that this strategy can assign accurate semantic labels to most channels. Fur-
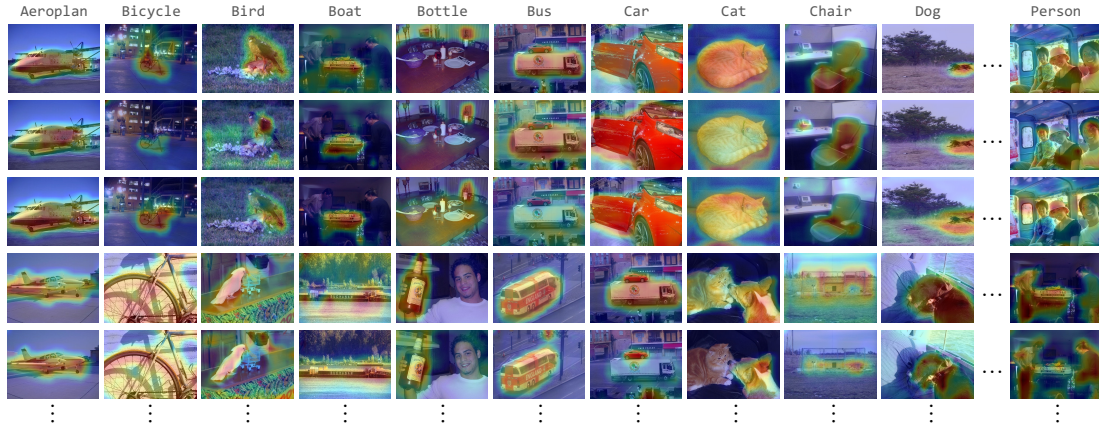
Figure 7. Visualization of part of the the Semantic Memory. Each image denotes one channel. For better visualization, we run inference on unlabeled training data, so that we can put the corresponding raw image under each channel.

Backbone	CCVC [46]	DGCL [44]	UniMatch [50]	Ours
R101+DeepLabV3+	74.40	77.14	77.19	73.70
SegFormer-B4	71.07↓	76.33↓	76.28↓	77.92↑
SegFormer-B5	73.77↓	76.72↓	76.56↓	78.41↑

Table 6. Ablations on different Segmentation Backbones on *original* PASCAL VOC 2012 dataset - 1/8 (183).

thermore, the examples in Figure 6 indicate that the proposed strategy may add similar features from other classes, *e.g.*, some channels of the `boat` are added to some similar classes, `aeroplane`, `bus`, *etc.*, which offers richer features as references to reform other classes.

Why tailored for transformer-based method? It might be argued that our improvements are due to a stronger backbone, SegFormer [48]. Considering this concern, we further carry out a comparison with the previous SOTAs with transformer backbones. As shown in Table 6, when switching to SegFormer-B4/B5, previous methods all have performance drops, since the training of transformer requires more annotated data to avoid over-fitting [38]. However, our *AllSpark* also exhibits poorer performance with ResNet101 and DeepLabV3+. This can be attributed to the limited receptive fields of CNNs. As shown in Fig. 8, when analyzing large objects (*e.g.*, car), the attention maps often prioritize a few salient local parts (*e.g.*, wheels). Consequently, when these features processed by *AllSpark*, there is a risk of erroneously querying channels that correspond to similar parts of different objects (*e.g.*, wheels of an aeroplane landing gear), leading to unnecessary noise and performance decay.

5. Conclusions

We notice the separate training approach in current SSSS frameworks will cause the dominance of labeled data and
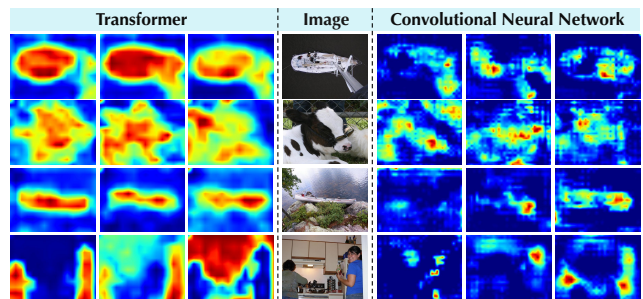

Figure 8. Comparison of Transformer and CNN feature channels.

thus lead to sub-optimal solutions. This separation hinders the effective integration of ViT backbones with SSSS frameworks. We propose *AllSpark* to alleviate this issue by directly reborning the labeled features with the unlabeled features. To further augment the unlabeled feature space, we introduce a Semantic Memory (S-Mem) to store previous unlabeled features from different classes. Then, we can leverage these comprehensive features from all classes to reconstruct more accurate labeled features. To establish the class-balanced S-Mem, we propose Channel Semantic Grouping strategy, which assigns semantic information to each channel based on its similarity with the probability map. We present that equipped with the proposed *AllSpark*, a naive pseudo-labeling scheme can outperform the SOTAs. Finally, *AllSpark* improves previous results remarkably on three well-established benchmarks.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 32, 2019. 1

[2] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. In *ICCV*, 2023. 2

[3] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, pages 357–366, 2021. 2

[4] 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. *IEEE TPAMI*, 40(4):834–848, 2017. 2

[5] 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. 1, 2, 6

[6] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*, pages 2613–2622, 2021. 1, 2, 3, 5, 6

[7] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 34:17864–17875, 2021. 2

[8] 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*, pages 3213–3223, 2016. 2, 5

[9] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *ECCV*, pages 74–92. Springer, 2022. 2, 4

[10] 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. *ICLR*, 2021. 2, 3

[11] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111:98–136, 2015. 2, 5

[12] Geoff French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, high-dimensional perturbations. *BMVC*, 2020. 6

[13] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *NeurIPS*, 34: 15908–15919, 2021. 2

[14] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 international conference on computer vision*, pages 991–998. IEEE, 2011. 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[16] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *NeurIPS*, 34:22106–22118, 2021. 6

[17] Huimin Huang, Shiao Xie, Lanfen Lin, Ruofeng Tong, Yen-Wei Chen, Yuexiang Li, Hong Wang, Yawen Huang, and Yefeng Zheng. Semicvt: Semi-supervised convolutional vision transformer for semantic segmentation. In *CVPR*, pages 11340–11349, 2023. 1, 3, 6

[18] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *CVPR*, pages 2989–2998, 2023. 2

[19] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. In *ICCV*, pages 752–761, 2023. 2

[20] Ying Jin, Jiaqi Wang, and Dahua Lin. Semi-supervised semantic segmentation via gentle teaching assistant. *NeurIPS*, 35:2803–2816, 2022. 1, 3, 6

[21] 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. 2

[22] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML, Workshops*, page 896, 2013. 1

[23] Peixia Li, Pulak Purkait, Thalaiyasingam Ajanthan, Majid Abdolshah, Ravi Garg, Hisham Husain, Chenchen Xu, Stephen Gould, Wanli Ouyang, and Anton van den Hengel. Semi-supervised semantic segmentation under label noise via diverse learning groups. In *ICCV*, pages 1229–1238, 2023. 1, 2, 3, 6

[24] Shuo Li, Yue He, Weiming Zhang, Wei Zhang, Xiao Tan, Junyu Han, Errui Ding, and Jingdong Wang. Cfcg: Semi-supervised semantic segmentation via cross-fusion and contour guidance supervision. In *ICCV*, pages 16348–16358, 2023. 1, 6

[25] Yijiang Li, Xinjiang Wang, Lihe Yang, Litong Feng, Wayne Zhang, and Ying Gao. Diverse cotraining makes strong semi-supervised segmentor. In *ICCV*, 2023. 3

[26] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Logic-induced diagnostic reasoning for semi-supervised semantic segmentation. In *ICCV*, pages 16197–16208, 2023. 1, 6

[27] 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. 2, 5

[28] Yuyuan Liu, Yu Tian, Yuanhong Chen, Fengbei Liu, Vasileios Belagiannis, and Gustavo Carneiro. Perturbed and strict mean teachers for semi-supervised semantic segmentation. In *CVPR*, pages 4258–4267, 2022. 6

[29] 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. 2

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1, 2

[31] Jie Ma, Chuan Wang, Yang Liu, Liang Lin, and Guanbin Li. Enhanced soft label for semi-supervised semantic segmentation. In *ICCV*, pages 1185–1195, 2023. 1, 6

[32] Pengchong Qiao, Zhidan Wei, Yu Wang, Zhennan Wang, Guoli Song, Fan Xu, Xiangyang Ji, Chang Liu, and Jie Chen. Fuzzy positive learning for semi-supervised semantic segmentation. In *CVPR*, pages 15465–15474, 2023. 6

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 1, 2

[34] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 33:596–608, 2020. 1

[35] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 2

[36] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, 2017. 1, 3

[37] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[38] Dusan Varis and Ondřej Bojar. Sequence length is a domain: Length-based overfitting in transformer models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8246–8257, 2021. 8

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

[40] Haonan Wang and Xiaomeng Li. Towards generic semi-supervised framework for volumetric medical image segmentation. In *NeurIPS*, 2023. 1

[41] Haonan Wang and Xiaomeng Li. Dhc: Dual-debiased heterogeneous co-training framework for class-imbalanced semi-supervised medical image segmentation. In *MICCAI*, pages 582–591. Springer, 2023. 1

[42] Haonan Wang, Peng Cao, Jiaqi Wang, and Osmar R Zaiane. Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer. In *AAAI*, pages 2441–2449, 2022. 2, 4

[43] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 2

[44] Xiaoyang Wang, Bingfeng Zhang, Limin Yu, and Jimin Xiao. Hunting sparsity: Density-guided contrastive learning for semi-supervised semantic segmentation. In *CVPR*, pages 3114–3123, 2023. 1, 3, 6, 8

[45] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *CVPR*, pages 4248–4257, 2022. 1, 3, 5, 6

[46] Zicheng Wang, Zhen Zhao, Xiaoxia Xing, Dong Xu, Xiangyu Kong, and Luping Zhou. Conflict-based cross-view consistency for semi-supervised semantic segmentation. In *CVPR*, pages 19585–19595, 2023. 2, 3, 6, 8

[47] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31, 2021. 2

[48] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34:12077–12090, 2021. 1, 2, 6, 7, 8

[49] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *CVPR*, pages 4268–4277, 2022. 6

[50] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In *CVPR*, pages 7236–7246, 2023. 1, 2, 3, 6, 8

[51] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, pages 10819–10829, 2022. 2

[52] Jianlong Yuan, Jinchao Ge, Zhibin Wang, and Yifan Liu. Semi-supervised semantic segmentation with mutual knowledge distillation. In *ACM MM*, pages 5436–5444, 2023. 6

[53] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, pages 558–567, 2021. 2

[54] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *NeurIPS*, 34:18408–18419, 2021. 1

[55] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 2

[56] Zhen Zhao, Lihe Yang, Sifan Long, Jimin Pi, Luping Zhou, and Jingdong Wang. Augmentation matters: A simple-yet-effective approach to semi-supervised semantic segmentation. In *CVPR*, pages 11350–11359, 2023. 1, 3, 6

[57] 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. 2

[58] Yuanyi Zhong, Bodi Yuan, Hong Wu, Zhiqiang Yuan, Jian Peng, and Yu-Xiong Wang. Pixel contrastive-consistent semi-supervised semantic segmentation. In *ICCV*, pages 7273–7282, 2021. 6

[59] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. *ICLR*, 2020. 1, 6