UniMatch V2: Pushing the Limit of Semi-Supervised Semantic Segmentation

Lihe Yang, Zhen Zhao†, and Hengshuang Zhao†

Abstract—Semi-supervised semantic segmentation (SSS) aims at learning rich visual knowledge from cheap unlabeled images to enhance semantic segmentation capability. Among recent works, UniMatch [1] improves its precedents tremendously by amplifying the practice of weak-to-strong consistency regularization. Subsequent works typically follow similar pipelines and propose various delicate designs. Despite the achieved progress, strangely, even in this flourishing era of numerous powerful vision models, almost all SSS works are still sticking to 1) using outdated ResNet encoders with small-scale ImageNet-1K pre-training, and 2) evaluation on simple Pascal and Cityscapes datasets. In this work, we argue that, it is necessary to switch the baseline of SSS from ResNet-based encoders to more capable ViT-based encoders (*e.g.*, DINOv2) that are pre-trained on massive data. A simple update on the encoder (even using 2× fewer parameters) can bring more significant improvement than careful method designs. Built on this competitive baseline, we present our upgraded and simplified UniMatch V2, inheriting the core spirit of weak-to-strong consistency from V1, but requiring less training cost and providing consistently better results. Additionally, witnessing the gradually saturated performance on Pascal and Cityscapes, we appeal that we should focus on more challenging benchmarks with complex taxonomy, such as ADE20K and COCO datasets. Code, models, and *logs of all reported values*, are available at https://github.com/LiheYoung/UniMatch-V2.

Index Terms—Semi-Supervised Learning, Semantic Segmentation, Weak-to-Strong Consistency, Vision Transformer.

1 Introduction

Semantic segmentation [2], [3], [4], [5] plays a fundamental role in scene understanding by providing pixel-level class predictions. However, substantial dense annotations are required to learn a capable semantic segmentation model. For example, it takes around 1.5 hours to label a single image on Cityscapes [6] with merely 19 classes. This limitation greatly hinders the deployment of advanced models in critical applications without sufficient annotations. Therefore, to alleviate the burden of human annotators and decrease annotation costs, semi-supervised semantic segmentation (SSS) is attracting increasing attention. SSS aims to train a model with a small portion of labeled images and take full advantage of more unlabeled images. One of the most representative works recently is Segment Anything [7]. It designs a semi-supervised data engine to gradually expand from small-scale human labels to automatically produced large-scale pseudo labels. Such methodology is universal and can be applied to many scenarios [8]. In this work, we especially focus on the task of semi-supervised semantic segmentation, which has been widely studied in recent years, covering extensive fields like natural image understanding [9], [10], [11], [12], medical image analysis [13], [14], [15], [16], remote sensing interpretation [17], [18], [19], *etc.*.

The core problem in SSS is how to effectively leverage unlabeled images. Existing works [1], [10], [20] mostly follow the methodology of pseudo labeling (also called self-training) [21], [22]. The model first acquires initial semantic segmentation ability from labeled images, and then assigns pseudo labels (*i.e.*, model predictions) to unlabeled images to expand available training samples. Such pseudo-labeling pipeline can be carried out either offline (*i.e.*, multiple stages) or online (*i.e.*, end-to-end). In an offline pipeline [20], the pseudo-labeling step is conducted only when the model has been sufficiently trained on labeled images. In contrast, for an online pipeline [21], the model predicts pseudo labels in each training iteration for the sampled unlabeled batch. From the very start of training, the model is jointly optimized on manually labeled images and pseudo-labeled images. Both offline and online roadmaps have witnessed

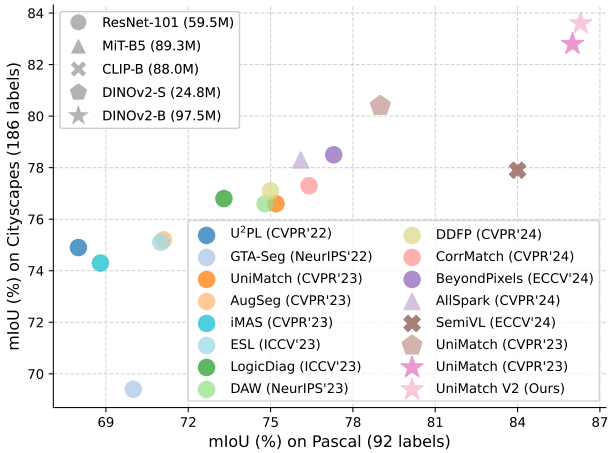


Fig. 1: Performance of various methods across different pre-trained encoders **(upper-left legend)**. Under the ResNet-101 backbone, previous works struggle to further improve the best results. But with a simple update on the backbone (ResNet-101 → DINOv2-S → DINOv2-B) while keeping the method unchanged, the UniMatch [1] performance is boosted significantly.

- *Lihe Yang and Hengshuang Zhao are with The University of Hong Kong. Email: lihe.yang.cs@gmail.com, hszhao@cs.hku.hk.*
- *Zhen Zhao is with Shanghai AI Laboratory. Email: zhaozhen@pjlab.org.cn*
- *†Corresponding authors: Zhen Zhao and Hengshuang Zhao.*

great development in the past few years.

Dating back to three years ago, ST++ [20] demonstrates that, a plain offline self-training pipeline [21] is indeed superior to previous online ones [23], [24], as long as injecting appropriate strong data augmentations to unlabeled images. Although such an offline strategy can ensure the quality of pseudo labels, it is not elegant enough, requiring three separate stages. In view of this, UniMatch [1] revisits the weak-to-strong consistency regularization, which is simplified and popularized by FixMatch [25] originally in semi-supervised classification. As an elegant online self-training framework, FixMatch estimates pseudo labels on weakly-augmented (*e.g.*, cropped) clean images and uses these labels to supervise the training of corresponding strongly-augmented (*e.g.*, color-jittered) images. To select reliable pseudo labels to learn, it pre-defines a confidence threshold and excludes model predictions not satisfying this criterion. Despite its simplicity and being proposed five years ago, UniMatch shows that, if equipped with strong spatial augmentations (*i.e.*, CutMix [26]), FixMatch is still a highly competitive baseline in SSS. It significantly outperforms all previous delicately designed methods before 2023.

FixMatch [25] harnesses rich visual knowledge by training on challenging strongly-augmented unlabeled images. However, the strong augmentations are constrained in the input space, *i.e.*, merely applying color and spatial distortions to raw images. This prohibits the model from pursuing invariant representations under a broader augmentation space. Thus, to further promote the spirit of weak-to-strong consistency in FixMatch, UniMatch [1] employs an additional feature-level augmentation stream as a supplement to the input-level stream. It finds a simplest channel-wise Dropout [27] on intermediate features works the most effectively. Moreover, to fully explore the original input-level augmentation space, it designs a dual-stream augmentation strategy at the input level. Two strongly-augmented images are jointly sampled from their shared weakly-augmented version through a random data augmentation pool. They are passed into the model in two parallel streams for training. With the two key practices (feature-level augmentation and dual-stream augmentation), UniMatch further improves the FixMatch performance remarkably. Due to the great simplicity and easily reproduced strong results, many subsequent works in SSS build their framework on UniMatch [28], [29], [30], [31] directly or on the more basic FixMatch [12], [32] reproduced by UniMatch.

However, after checking recent works in SSS, we noticed that their methods are becoming increasingly sophisticated. More importantly, even with these carefully designed modules, performance is usually only boosted by nearly 0.5% on datasets like Pascal and Cityscapes. We can expect that, if we continue going this way, future works in this field will have much more difficulty in improving the current state-of-the-art (SOTA) results. As a result, they will have trouble in publishing their works just due to being "not SOTA". This will greatly hinder new ideas or new frameworks to flourish. As a fundamental research topic, development in SSS can provide valuable insights and guidelines for real-world computer vision (CV) applications on how to utilize unlabeled data effectively [7], [8], [33], [34]. Therefore, we believe it is urgent to re-explore new meaningful roadmaps for future research on SSS.

Reviewing the development of SSS, numerous works have been published by "designing novel methods". In earlier years, new methods can bring a remarkable improvement of more than 5% [10]. However, recently, we no longer observe such significant advances. Most works only improve their precedents very marginally (around 0.5%). It indicates that we have almost arrived at the boundary of the modeling capability of current models or the potential upper bound of evaluated benchmarks. Meantime, jumping out of our narrow SSS field, in the past few years, the wider CV community has witnessed tremendous progress in 1) new model architectures, *e.g.*, vision transformers [35], [36], [37], better convolutional networks [38], [39], 2) better pre-training strategies [40], [41], [42], especially vision-alone self-supervised learning methods [43], [44], [45], [46], [47], and 3) leveraging ultra-massive data (over 100M) for pre-training [40], [47], [48], [49]. Despite these exciting progress, pitifully, none of them have been well integrated into our SSS field. Recent SSS works are still sticking to the outdated ImageNet-1K [50] pre-trained ResNet encoders [51]. To some extent, it can be understood because most previous SSS works have established the comparison baselines using the same encoder. It will be risky and costly to re-benchmark existing methods with new architectures. However, to make the SSS works have a broader impact, we believe it is worthy and urgent to do so, because there is no guarantee that insights obtained from these outdated encoders with small-scale pre-training can be safely transferred to modern architectures with large-scale pre-training.

Among latest SSS works, there are two exceptions discarding the ResNet encoders. One is SemiVL [31], building on the CLIP-ViT-B model [40], and the other is AllSpark [32], using the MiT-B5 [52] pre-trained encoder. Despite taking a step further, their used encoders are not powerful enough. And their experiments are not thorough enough to cover all datasets and fine-tuning strategies.

In this work, we aim at *re-building a comprehensive new benchmark* for semi-supervised semantic segmentation with the most capable pre-trained model DINOv2 [47]. Thanks to its large-scale curated data and advanced training strategies, DINOv2 exhibits superior performance in widespread scenarios, *e.g.*, classification, dense matching. As shown in Figure 1, without bells and whistles, a simple update on the pre-trained encoder from ResNet-101 to DINOv2-S (even 2$\times$ fewer parameters) boosts the UniMatch [1] performance by over 3% on Pascal and 4% on Cityscapes. And DINOv2-B is much stronger than other encoders of similar scales, such as CLIP-B [40] and MiT-B5 [52]. Impressed by the results, we re-conduct all core experiments of UniMatch V1 [1] and its baseline FixMatch [25] with DINOv2.

Built on this strong baseline (UniMatch + DINOv2), we further present our *upgraded and simplified UniMatch V2*. It inherits the core spirit of weak-to-strong consistency regularization from V1. But differently, V2 uses fewer strongly-augmented streams for learning. It fuses the feature-level Dropout and input-level augmentations into a single stream. Moreover, to fully explore the joint augmentation space, we design a Complementary Dropout at the feature level. It decomposes the feature maps along the channel dimension into two disjoint and complementary sets. The two non-

overlapping feature sets can be considered as two different yet meaningful views of an image. For example, one set of features may be sensitive to textures, while another set is responsible for the structure information. We then forward these two complementary features into the shared decoder for dual-stream learning. Compared with the dual-stream practice in V1 (twice random image augmentations), our Complementary Dropout can produce better dual views, which proves more effective in practice.

This work greatly updates our CVPR'23 work UniMatch V1 [1], with **multiple new contributions**: **(1)** We re-evaluate UniMatch V1 as well as its baseline FixMatch [25] (two most widely adopted methods in SSS) with the most capable vision foundation model DINOv2 [47]. **(2)** We simplify V1 by fusing its feature-level and input-level augmentations into a single learnable stream. Additionally, we propose a Complementary Dropout to fully harness dual-stream training. With the two upgrades, V2 shows more efficient training and better performance than V1. **(3)** We conduct extensive experiments, including but not limited to i) four popular semantic segmentation datasets, ii) a wide range of vision encoders: ResNet [51], DINOv2 [47], SAM [7], MiT [52], BEiT [45], iii) fine-tuning *vs.* freezing the pre-trained encoder, iv) comprehensive ablation studies on different frameworks and hyper-parameters across abundant settings, v) real-world large-scale SSS setting with considerable labeled images and much more unlabeled images, and vi) broader semi-supervised scenarios, *e.g.*, remote sensing changing detection and image classification.

2 Related Work

2.1 Semi-Supervised Learning

Semi-supervised learning (SSL) [53], [54], [55] studies how to better utilize unlabeled images. It is a fundamental and long-standing problem in machine learning, with extensive applications ranging from classical image classification [21] to modern CV and NLP foundation models [7], [8], [56]. Since the rise of deep neural networks [57], [58], [59], it has earned increasing attention. The reason behind this is that deep models are capable of fitting various training samples and then generalizing, but they are extremely hungry for labeled data, which are usually costly to acquire. The lack of training samples will greatly limit the generalization ability of modern networks. Therefore, researchers resort to cheap unlabeled data that widely exist and are effortless to collect. They can play a valuable role in increasing the data coverage and enhancing the model transferring capability.

To effectively incorporate unlabeled data, there are two mainstream methods. One is called *entropy minimization* [22], [60], [61], [62], [63], popularized by a straightforward self-training pipeline [21]. Typically, it first trains a teacher model on initial labeled data. And then this teacher model predicts pseudo labels on additional unlabeled data. The predicted logits are usually sharpened or post-processed as one-hot pseudo labels in classification. These hard labels with minimized entropy can serve as supervision signals to re-train a student model on the unlabeled data. To better use this pipeline, one key insight from Noisy Student [22] is that we should inject strong data augmentations and model augmentations when re-training the student model to increase its learning difficulty.

Despite its effectiveness, the offline self-training pipeline has been rarely used, due to its inconvenient three stages. Many recent works prefer the end-to-end pseudo-labeling frameworks powered by *consistency regularization* [25], [64], [65], [66], [67]. The core idea is to align the poisoned predictions with more accurate predictions. From the model aspect, Mean Teacher [64] maintains an EMA teacher model to produce better pseudo labels for the student. From the data aspect, FixMatch [25] uses the high-quality prediction on a clean image to supervise the training of corresponding strongly-augmented hard image. Due to its simplicity and efficacy, numerous subsequent works [68], [69], [70], [71], [72], [73], [74], [75] build on it. Among them, FlexMatch [76] replaces the global confidence threshold with class-aware thresholds by considering the class-wise learning status. DST [77] decouples the prediction and learning of pseudo labels with separate heads to suppress model noise. Beyond semantic-level consistency, SimMatch [78] applies an auxiliary regularization on instance-level relationships. To safely use all unlabeled data, ShrinkMatch [79] excludes confusion classes for uncertain samples to learn.

Our work inherits the weak-to-strong consistency regularization from FixMatch. Differently, we propose a dual-stream Complementary Dropout to craft better augmentation space. Besides, we focus on semi-supervised semantic segmentation, which is more laborious to acquire labeled data than the classification task of FixMatch.

2.2 Semi-Supervised Semantic Segmentation

Semi-supervised semantic segmentation (SSS) is an essential subfield of SSL, with extensive applications in scene understanding [9], [10], [11], [12], medical image analysis [13], [14], [15], [16], and remote sensing interpretation [17], [18], [19]. Earlier works [9], [80] make some pioneering efforts in borrowing the methodology of GANs [81] to SSS. They train the segmentation model as a generator to produce predictions that can fool the network used to discriminate pseudo labels from manual labels. Despite inspiring, such frameworks are very hard to train. Later works [24], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95] till now, mostly follow the progress of SSL, proposing simpler yet useful designs from the perspectives of *entropy minimization* and *consistency regularization*. These two principles are usually incorporated together.

During this trend, French *et al.* [23] highlight the necessity of injecting Cutout [96] and CutMix [26] into unlabeled images to challenge the model to learn. Then CPS [10] reveals that CutMix can work more effectively with a co-training framework [97], [98], [99]. AEL [100] further designs an adaptive CutMix to bias the sampled data towards under-performing classes. These three works use an online pseudo-labeling framework. Their pseudo label quality cannot be guaranteed at the early training stages. In contrast, ST++ [20] and SimpleBaseline [101] adopt a three-stage self-training pipeline, pseudo labeling only when the model has been fully trained on labeled data. Despite promising results, they are not elegant enough.

To simplify the SSS methods and disclose what really matters, UniMatch [1], AugSeg [102], and iMAS [103] build

simple yet effective frameworks based on the methodology of FixMatch [25]. They all emphasize the critical role of strong data augmentations on unlabeled images, such as color jittering and CutMix. They simply pre-define a confidence threshold to discard noisy pseudo labels, which proves very useful. In detailed designs, UniMatch [1] reveals the benefits of dual-stream augmentations and feature-level augmentations. AugSeg [102] increases the randomness in RandAugment [104] for richer data augmentations. And iMAS [103] employs adaptive augmentations and supervisions based on the model state.

Due to their simplicity and easily reproduced strong results, subsequent works mostly follow them, or the more basic FixMatch framework reproduced by UniMatch. Among the latest works, CorrMatch [12] leverages correlation maps to propagate and polish pseudo labels. AllSpark [32], different from most works centered on unlabeled data, points out the value of labeled features. Most recently, SemiVL [31] combines a CLIP [40] encoder into SSS via a spatial fine-tuning module and a language-aware decoder. Besides, BeyondPixels [29] finds it is helpful to additionally train a patch-level classifier for contexture information.

Despite the progress, these works do not adopt a powerful vision encoder, mostly sticking to the outdated ResNet encoder. We aim to re-benchmark all existing settings in SSS with the most capable DINOv2 encoder [47]. Moreover, we present a simplified yet stronger UniMatch V2 framework based on this modern architecture.

3 Methodology

Our proposed UniMatch V2 in this work, and our precedent work UniMatch V1 [1] are both based on FixMatch [25]. Thus we will primarily introduce this legacy framework in Section 3.1. Then, we will go through our prior UniMatch V1 framework in Section 3.2, analyzing its advantages and limitations. Lastly, in Section 3.3, we will present our simplified UniMatch V2 framework, which is more efficient and more capable than V1.

3.1 Preliminaries

Generally, semi-supervised semantic segmentation (SSS) involves two datasets: a labeled dataset $\mathcal{D}^l = \{(x_i^l, y_i^l)\}$ and an unlabeled dataset $\mathcal{D}^u = \{x_i^u\}$, where x_i is the i-th image and y_i is its ground-truth semantic mask. In most cases, $\mathcal{D}^u$ is at a much larger scale than $\mathcal{D}^l$, *e.g.*, 10× more images. The model learns initial visual knowledge from the small portion of labeled images, and then uses such knowledge to harvest the value of abundant unlabeled images.

To take full advantage of unlabeled data, FixMatch [25] (Figure 2a) adopts a weak-to-strong consistency regularization framework [65], [66], [67], which is trained in an end-to-end manner. Concretely, each mini-batch is composed of B^l labeled images and B^u unlabeled images. On the labeled images, the model is supervised by manually provided labels. This loss $\mathcal{L}^l$ can be formulated as:

$$\mathcal{L}^l = \frac{1}{B^l} \sum_{i=1}^{B^l} \mathrm{H}(p_i^l, y_i^l), \tag{1}$$

where p_i^l is the model prediction on the i-th labeled image, and y_i^l is its corresponding ground-truth mask. H is the widely used hard cross-entropy loss.

On the unlabeled images, the model first assigns pseudo labels (*i.e.*, model predictions) to them and then learns in a self-teaching manner. Most importantly, in a weak-to-strong consistency regularization pipeline, the model predicts pseudo labels on the *weakly-augmented image* x^w, but learns (*i.e.*, trains) on its *strongly-augmented version* x^s. The underlying logic of this asymmetric practice is that 1) the model produces higher-quality pseudo labels on the clean image x^w, but 2) directly training on x^w will incur a minimal loss (little information), while x^s is more appropriate for training since it can challenge the model to seek invariance under strong augmentations.

The x^w is generated by feeding the original unlabeled image x^w into a weak data augmentation pool $\mathcal{A}^w$, including basic image operators like cropping, resizing, and horizontal flipping. The x^s is further yielded from x^w through a strong data augmentation pool $\mathcal{A}^s$. A^s consists of intensive color distortions and layout changes (*i.e.*, CutMix [26]). Formally, this data pre-processing follows:

$$x^w = \mathcal{A}^w(x^u), \quad x^s = \mathcal{A}^s(x^w). \tag{2}$$

The model f makes predictions on the two versions:

$$p^w = f(x^w), \quad p^s = f(x^s). \tag{3}$$

As aforementioned, p^w is considered as the pseudo label. In practice, the softmax output p^w is further post-processed by an $\arg\max$ operator to become a hard one-hot label $\hat{p}^w$. It then supervises p^s to train on the unlabeled data:

$$\mathcal{L}^u = \frac{1}{B^u} \sum_{i=1}^{B^u} \mathbb{1}(\max(p_i^w) \geq \tau) \mathrm{H}(p_i^s, \hat{p}_i^w), \tag{4}$$

where $\mathbb{1}(\max(p_i^w) \geq \tau)$ is a special design in FixMatch to alleviate the negative effect of noisy pseudo labels. It predefines a confidence threshold τ (*e.g.*, 0.95). Pseudo labels not satisfying this threshold will be excluded from training. This mechanism ensures that, in early training iterations, the model is primarily optimized on high-quality manually labeled images, and then the training is progressively expanded to confidently pseudo-labeled images.

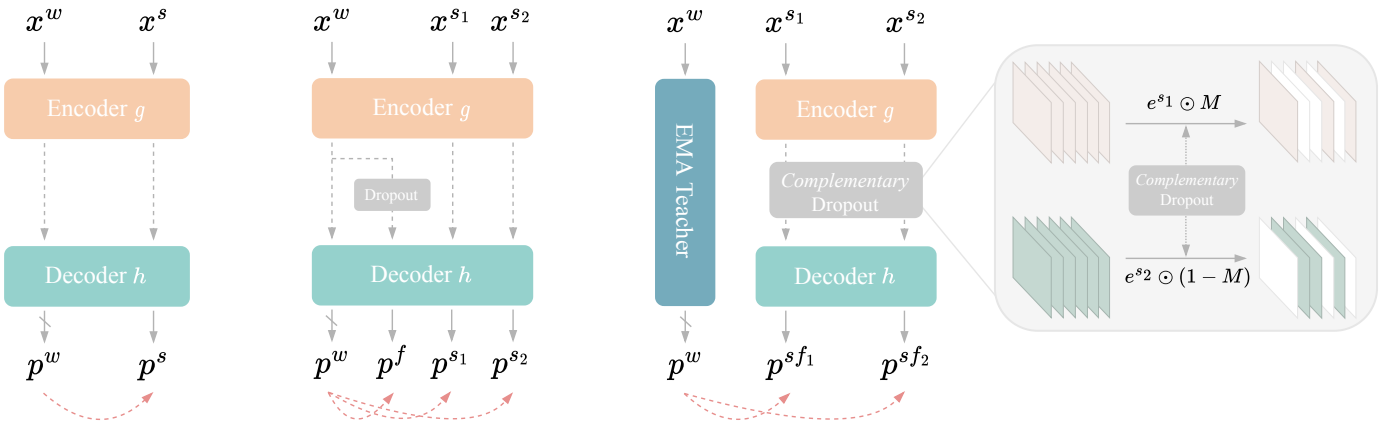
Finally, the joint loss for a mini-match is a simple interpolation of the labeled and unlabeled loss:

$$\mathcal{L} = \mathcal{L}^l + \lambda \mathcal{L}^u, \tag{5}$$

where λ balances the effect of the unlabeled data. We simply set it as 1 in all our experiments.

3.2 UniMatch V1: Unified Dual-Stream Augmentations

Motivated by the impressive results of FixMatch when reproduced in SSS, UniMatch V1 [1] (Figure 2b) aims at further strengthening its weak-to-strong consistency regularization from two perspectives, namely unified image-level and feature-level augmentations (Section 3.2.1) and dual-stream augmentations (Section 3.2.2).



(a) FixMatch [25] **(b)** UniMatch V1 [1] **(c)** Our UniMatch V2, unified augmentations and complementary Dropout

Fig. 2: Illustration of the evolution from FixMatch (a) to our prior UniMatch V1 (b), and to our current UniMatch V2 (c). FixMatch uses the prediction of a weakly-augmented image to supervise the corresponding strongly-augmented image. Based on FixMatch, UniMatch V1 brings a separate feature-level augmentation (*i.e.*, Dropout) stream and an additional image-level augmentation stream. Our UniMatch V2, simpler and stronger than V1, unifies image-level and feature-level augmentations into a single stream (Section 3.3.1) and presents complementary Dropout to craft better dual views (Section 3.3.2).

3.2.1 Unified Augmentations of Image and Feature Levels

FixMatch has achieved great success [105], [106], [107], [108], [109], [110] by enforcing invariant model predictions under strong data augmentations $\mathcal{A}^s$. However, optimal $\mathcal{A}^s$ is not trivial to obtain, especially for images of specific domains, such as medical images and aerial images. It mostly requires domain experts to dig out the appropriate $\mathcal{A}^s$. This may limit a broader impact of our semi-supervised algorithms. More importantly, the augmentations (*e.g.*, color jittering, CutMix) in FixMatch are all constrained in the image space, hindering the model from exploring more robust invariance under a broader augmentation space. Especially in this era of foundation models, some pre-trained encoders have been proven to be highly robust to color distortions [111]. Thus, such image-only augmentations may be inadequate to fully unleash the potential of unlabeled images.

To this end, UniMatch V1 proposes to construct unified augmentations at both image and feature levels. Apart from the learnable stream of the strongly-augmented *image* x^s, it maintains an additional stream of the strongly-augmented *intermediate features*. Different from prior works [112], [113] that combine all types of augmentations in a single stream, V1 reveals that it is better to disentangle different levels of augmentations into separate streams to avoid a single stream being excessively hard to learn. So V1 injects feature-level strong augmentations to weakly-augmented images x^w, rather than x^s. Formally, suppose a semantic segmentation model f is composed of an encoder g (*e.g.*, ResNet [51], DINOv2 [47]) and a decoder h (*e.g.*, ASPP [3], DPT [114]). Then, this process can be formulated as:

$$e^w = g(x^w), \tag{6}$$
$$p^f = h(\mathcal{F}(e^w)), \tag{7}$$

where e^w is the extracted clean intermediate features of the weakly-augmented image x^w. $\mathcal{F}$ is feature-level augmentations, such as Dropout [27], adding uniform noise, or VAT [115]. And p^f is the decoder (h) output (*i.e.*, model prediction) of the strongly-augmented features.

After incorporating this additional learnable stream on p^f, the overall unlabeled loss is updated as:

$$\mathcal{L}^u = \frac{1}{2B^u} \sum_{i=1}^{B^u} \mathbb{1}(\max(p_i^w) \geq \tau)\big(\mathrm{H}(p_i^s, \hat{p}_i^w) + \mathrm{H}(p_i^f, \hat{p}_i^w)\big), \tag{8}$$

where we assign equal loss weights for the image-level and feature-level learnable streams.

Through extensive ablation studies, V1 reveals a simple channel-wise Dropout (`nn.Dropout2d(0.5)` in PyTorch) works pretty well as the feature augmentation. It randomly selects half of the feature maps along the channel dimension and masks them out with zero value. We find that there is no need to adopt the computationally intensive VAT practice [113], [115] to perturb features. It is also worth noting that, in practice, a pre-trained encoder mostly outputs multi-level intermediate features for the input of the subsequent decoder. For example, the ASPP decoder [3] takes both first-stage and final-stage features of the ResNet encoder. In such cases, we apply the feature augmentations to each feature volume independently.

3.2.2 Dual-Stream Augmentations

The unified augmentations above successfully strengthen FixMatch by expanding the augmentation space. It enforces the model to seek robust representations under richer distortions, which is key to stronger generalization ability. Apart from this modification, UniMatch V1 further proposes to explore the original input-level augmentation space more thoroughly. To this end, it designs a dual-stream augmentation strategy. This strategy is motivated by the multi-view learning techniques in self-supervised learning and semi-supervised classification. For example, SwAV [116] uses a multi-crop technique to divide an image into multiple views of different resolutions, and then optimizes the local-to-global consistency among them. Similarly, ReMixMatch [67] produces multiple strongly-augmented images for the model to learn jointly.

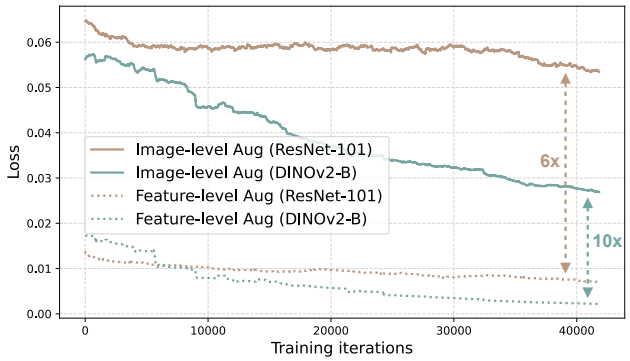


Fig. 3: Comparison between the loss scales under image-level augmentations (*e.g.*, color jittering, CutMix) and feature-level augmentation (*i.e.*, Dropout). Loss incurred by image augmentations is much larger than that of feature augmentation.

Concretely, in our UniMatch V1, we first obtains two strongly-augmented images (x^{s_1}, x^{s_2}) from their shared weakly-augmented version x^w:

$$x^{s_1} = \mathcal{A}^s(x^w), \quad x^{s_2} = \mathcal{A}^s(x^w), \tag{9}$$

where x^{s_1} and x^{s_2} are not equal, since the pre-defined strong data augmentation pool $\mathcal{A}^s$ are not deterministic. For example, the two versions may go through different types or strengths of color distortions. And their randomly selected CutMix regions may also be different.

We forward the two versions of images into the model in parellel. Their corresponding predictions p^{s_1} and p^{s_2} are jointly supervised by the high-quality prediction of their shared weakly-augmented version:

$$\mathcal{L}^u = \frac{1}{2B^u} \sum_{i=1}^{B^u} \mathbb{1}(\max(p_i^w) \geq \tau) \big(H(p_i^{s_1}, \hat{p}_i^w) + H(p_i^{s_2}, \hat{p}_i^w)\big) \tag{10}$$

We find in V1 that, despite the great simplicity, the dual-stream augmentations are highly beneficial to FixMatch. It is further validated that, the performance gain is non-trivial, not credited to a doubled unlabeled batch size. Compared with the single-stream augmentation, our design not only fully explores the original augmentation space, but also stabilizes the training. For instance, single-stream augmentation may produce excessively hard or easy images. In such cases, introducing an additional parallel stream for learning can serve as a valuable balance in model optimization (*i.e.*, gradient descent). Moreover, we conjecture that supervising two strong views with a shared weak view can be conceptually regarded as enforcing consistency between these two strong views. Thus, our dual-stream practice shares the core spirits of contrastive learning [43], [44], [117], which is able to learn discriminative representations and has been proved to be highly meaningful to our SSS task [118], [119]. We also adopt similar dual-stream methodology in our UniMatch V2, discussed next.

3.2.3 Summary and Discussion

Incorporating the above two designs of unified augmentations and dual-stream augmentations, there are three learnable streams in UniMatch V1, as well as one inference stream used to produce pseudo labels. The three streams enforce the model to keep consistent under multi-level random strong augmentations, greatly strengthening the weak-to-strong consistency regularization practice of FixMatch. The final loss on unlabeled data can be formulated as:

$$\mathcal{L}_{v_1}^u = \frac{1}{2B_u} \sum_{i=1}^{B_u} \mathbb{1}(\max(p_i^w) \geq \tau) \cdot \big(\mathrm{H}(p_i^f, \hat{p}_i^w) + \frac{1}{2}\big(\mathrm{H}(p_i^{s_1}, \hat{p}_i^w) + \mathrm{H}(p_i^{s_2}, \hat{p}_i^w)\big)\big). \tag{11}$$

It can be observed that we evenly accumulate losses of the dual-stream augmentations. We also directly set equal weights for the two losses of input-level and feature-level augmentations. For the universality, we avoid carefully fine-tuning such hyper-parameters for better results. We still strictly obey such principles in our V2.

With these effective designs, UniMatch V1 successfully pushes the SSS results to a new bar. It has become the new baseline for many subsequent works [28], [29], [30], [31]. However, it is still not optimal. For example, although there is only a single stream for final inference, three learnable streams are computationally heavy during training. Moreover, as exhibited in Figure 3, the feature-level augmentation stream incurs a much smaller loss than the image-level augmentation, *e.g.*, 6× smaller. When applying more advanced DINOv2 encoders, the feature-level loss will become even more negligible, contributing less to the final performance. Therefore, our UniMatch V2 aims to improve the training efficiency of V1, and further enhances its performance under capable vision foundation models.

Algorithm 1: Pseudocode of UniMatch V2 in a PyTorch style.

```
# f: network, composed of an encoder g and a decoder h
# f_ema: EMA teacher of f
# aug_w/aug_s: weak/strong image-level perturbations

# use binomial distribution to generate binary dropout masks
binomial = torch.distributions.binomial.Binomial(probs=0.5)

# cross-entropy loss function
criterion = torch.nn.CrossEntropyLoss()

for x_u in loader_u:
   # one weak view and two strong views as input
   x_w = aug_w(x_u)
   x_s1, x_s2 = aug_s(x_w), aug_s(x_w)

   # pseudo label obtained from weakly-augmented image
   pred_w = f_ema(x_w)
   mask_w = pred_w.argmax(dim=1).detach()

   # features (BxCxHxW) of dual strongly-augmented images
   feat_s = g(torch.cat((x_s1, x_s2)))

   # generate complementary channel-wise Dropout masks
   bs, dim = pred_w.shape[:2]
   mask_s1 = binomial.sample((bs, dim))
   mask_s2 = 1 - mask_s1
   mask = torch.cat((mask_s1, mask_s2)) * 2

   # perform Dropout on features
   feat_sf = feat_s * mask[..., None, None]

   # final decoder prediction after strong augmentations
   pred_sf = h(feat_sf)

   # loss from the dual streams
   loss_u = criterion(pred_sf, mask_w.repeat(2, 1, 1))
```

3.3 UniMatch V2: *Simpler* and *Stronger*

Primarily, from the architecture aspect, we construct our UniMatch V2 framework (Figure 2c) on the most capable DINOv2 encoder. Almost all previous works, including our V1, still blindly use the outdated ResNet encoders, just to more conveniently compare with existing works. However,

through our ablation studies (Table 9), the most lightweight DINOv2-Small encoder significantly outperforms the heaviest ResNet-152 encoder in our SSS task, using 3× fewer model parameters (24.8M *vs.* 78.6M). Therefore, we appeal that it is urgent to switch to these modern encoders for a broader impact of future SSS works. And we will discuss our technical designs in the context of DINOv2.

Technically, in V2, we aim to 1) reduce the number of trainable streams in V1 for improved training efficiency, 2) still maintain the core spirit of unified image-level and feature-level augmentations, and 3) achieve further better results than V1 under the modern encoders.

3.3.1 Single-Stream Unified Augmentations

As revealed in Figure 3, the loss incurred by purely feature-level augmentations is very marginal compared with the loss under input-level augmentations. The gap is even larger when updating the weak ResNet encoder to the powerful DINOv2 encoder. Through our experiments, we find there is no performance loss when removing the feature-level augmentation stream under DINOv2, *e.g.*, four out of five settings on ADE20K are even improved by removing it. However, we believe that enforcing the model to be resistant to multi-level augmentations is still beneficial. To this end, we propose to unify the input-level and feature-level augmentations into *a single stream*. This practice has three advantages: 1) the model can still pursue robust representation in a broad augmentation space, 2) there is no need to maintain separate streams for different forms of augmentations, thus the training efficiency is improved, and 3) DINOv2 is much more capable than previous ResNet, so it is beneficial to *stack* various augmentations to further challenge it to learn, rather than *decoupling* them as V1. The single-stream loss is formulated as:

$$p^{sf} = h(\mathcal{F}(g(x^s))), \tag{12}$$

$$\mathcal{L}^u = \frac{1}{B^u}\sum_{i=1}^{B^u} \mathbb{1}(\max(p_i^w) \geq \tau)\mathrm{H}(p_i^{sf}, \hat{p}_i^w). \tag{13}$$

3.3.2 Complementary Channel-Wise Dropout

Greatly impressed by the promising results of dual-stream augmentations in V1, we aim to amplify this methodology in our V2. Based on the above single-stream unified augmentations, an intuitive dual-stream practice is to randomly produce two strongly-augmented images, and then apply a random Dropout to their respective features for decoding. Despite promising, such a practice fails to fully decouple the dual streams for learning. Therefore, we further propose a *Complementary Channel-Wise Dropout* to acquire two disjoint and complementary sets of features for the dual streams. Concretely, given a feature map $e^{s_1} \in \mathbb{R}^{B\times C\times H\times W}$ from the x^{s_1} input stream, we randomly generate a dropout mask M of the same shape as e^{s_1}. M is a binary mask sampled from the binomial distribution (probability is 0.5), with half of its channels ($C/2$) all set as 1, and others as 0. With M, we can perform complementary channel-wise Dropout on features e^{s_1} and e^{s_2} (obtained from $g(x^{s_1})$ and $g(x^{s_2})$):

$$e^{s_1} \leftarrow e^{s_1} \odot M \times 2, \tag{14}$$

$$e^{s_2} \leftarrow e^{s_2} \odot (1-M) \times 2, \tag{15}$$

where the M and $1-M$ are two complementary dropout masks. And the last scaling factor "2" is to ensure the output expectation is of the same scale as normal features.

Note that e^{s_1} and e^{s_2} are extracted by a shared encoder. They share the same characteristics per channel. Therefore, the complementary Dropout masks M and $(1-M)$ will enable us to obtain two disjoint sets of features with disjoint meanings for the dual-stream learning. Moreover, although the Dropout is inserted at the intersection of the encoder and decoder, the gradient will be back-propagated to the encoder, making the entire model more robust.

Given the dual complementary streams of input-level and feature-level augmentations, the final unlabeled loss in UniMatch V2 is formulated as:

$$p^{sf_1} = h(e^{s_1}), \quad p^{sf_2} = h(e^{s_2}), \tag{16}$$

$$\mathcal{L}_{v_2}^u = \frac{1}{2B^u}\sum_{i=1}^{B^u} \mathbb{1}(\max(p_i^w) \geq \tau)\big(\mathrm{H}(p_i^{sf_1}, \hat{p}_i^w) + \mathrm{H}(p_i^{sf_2}, \hat{p}_i^w)\big) \tag{17}$$

Another minor modification on V1 is, we maintain an exponentially moving averaged teacher model [64], [102], [113], rather than using the student model itself, to produce stable and better pseudo labels. Teacher parameters θ_t are updated alongside the student parameters θ^s by:

$$\theta^t \leftarrow \gamma \times \theta^t + (1-\gamma) \times \theta^s, \tag{18}$$

where γ is dynamically set as $\min(1 - \frac{1}{\text{iter}+1}, 0.996)$.

A PyTorch-like pseudocode of our UniMatch V2 is provided in Algorithm 1. It is conceptually simple to implement yet meantime highly effective.

4 Experiment

In this paper, our primary goal is to thoroughly update the outdated ResNet encoders with the powerful DINOv2 encoders in semi-supervised semantic segmentation (SSS). Therefore, we conduct comprehensive experiments, covering all previously used datasets and protocols, as well as not yet explored but more challenging and practical datasets that we wish to promote. In addition to the achieved state-of-the-art (SOTA) results of our UniMatch V2, we also provide the results of our UniMatch V1, our baseline FixMatch, and the labeled-only results with DINOv2. Moreover, we compile extensive ablation studies across a wide range of settings to demonstrate the effectiveness of our method. Considering the limited computational resources of some academic groups and to better facilitate future research, we also re-benchmark all the results *under frozen pre-trained encoders* (less GPU memory, less training time).

Except the learning rate, which is necessary to re-explore an optimal value for the new encoder, we do not intensively fine-tune any hyper-parameters. We even keep the learning rate identical for all our explored settings. Therefore, despite our much stronger results than previous works, we believe there is still much room to further improve.

4.1 Datasets

We evaluate our UniMatch V2 on four popular benchmarks in semantic segmentation, *i.e.*, Pascal [126], Cityscapes [6], ADE20K [127], and COCO [128]. The first two datasets

Pascal SOTAs	Venue	Encoder	Ratio and absolute number of labeled images					#Params
			1/16 (92)	1/8 (183)	1/4 (366)	1/2 (732)	Full (1464)	
Labeled Only	-	RN-101	45.1	55.3	64.8	69.7	73.5	59.5M
ST++ [20]	CVPR'22	RN-101	65.2	71.0	74.6	77.3	79.1	59.5M
U^2PL [119]	CVPR'22	RN-101	68.0	69.2	73.7	76.2	79.5	59.5M
PS-MT [113]	CVPR'22	RN-101	65.8	69.6	76.6	78.4	80.0	59.5M
GTA-Seg [120]	NeurIPS'22	RN-101	70.0	73.2	75.6	78.4	80.5	59.5M
PCR [121]	NeurIPS'22	RN-101	70.1	74.7	77.2	78.5	80.7	59.5M
iMAS [103]	CVPR'23	RN-101	68.8	74.4	78.5	79.5	81.2	59.5M
AugSeg [102]	CVPR'23	RN-101	71.1	75.5	78.8	80.3	81.4	59.5M
UniMatch V1 [1]	CVPR'23	RN-101	75.2	77.2	78.8	79.9	81.2	59.5M
UniMatch V1 [1]	CVPR'23	CLIP-B	77.9	80.1	82.0	83.3	84.0	88.0M
Diverse CoT [122]	ICCV'23	RN-101	75.7	77.7	80.1	80.9	82.0	59.5M
ESL [123]	ICCV'23	RN-101	71.0	74.0	78.1	79.5	81.8	59.5M
LogicDiag [124]	ICCV'23	RN-101	73.3	76.7	77.9	79.4	-	59.5M
DAW [28]	NeurIPS'23	RN-101	74.8	77.4	79.5	80.6	81.5	59.5M
DDFP [125]	CVPR'24	RN-101	75.0	78.0	79.5	81.2	82.0	59.5M
CorrMatch [12]	CVPR'24	RN-101	76.4	78.5	79.4	80.6	81.8	59.5M
AllSpark [32]	CVPR'24	MiT-B5	76.1	78.4	79.8	80.8	82.1	89.3M
BeyondPixels [29]	ECCV'24	RN-101	77.3	78.6	79.8	80.8	81.7	59.5M
SemiVL [31]	ECCV'24	CLIP-B	84.0	85.6	86.0	86.7	87.3	88.0M
Labeled Only (85.0)	-	DINOv2-S	67.0	75.6	81.8	83.7	85.6	**24.8M**
UniMatch V2	Ours		79.0	85.5	85.9	86.7	87.8	
Labeled Only (86.5)	-	DINOv2-B	76.9	82.1	85.3	87.2	88.3	97.5M
UniMatch V2	Ours		**86.3**	**87.9**	**88.9**	**90.0**	**90.8**	

TABLE 1: Comparison with state-of-the-art methods on **Pascal** high-quality set. The number (*e.g.*, 85.0) next to "Labeled Only" denotes the fully-supervised result (1464 precisely labeled images + 9118 coarsely labeled images).

are widely used in SSS, but the last two are rarely evaluated, due to complex taxonomies. However, considering the saturated results on Pascal and Cityscapes, we believe we should pay more attention to ADE20K and COCO.

Pascal [126] (2012 version) contains 1464 images with high-quality semantic masks and 9118 images with less precise masks, spanning 21 classes. In this work, we select labeled images from the high-quality set, and treat all other images as unlabeled ones.

Cityscapes [6] is a classical urban scene dataset with 2975 images of 1024×2048 resolution, covering 19 classes. Although the class space is relatively small, the label quality is very high due to the high resolution, containing many thin objects, such as traffic lights.

ADE20K [127] is a rather challenging dataset composed of 150 classes. It includes 20210 labeled images. It is widely used in fully-supervised semantic segmentation, but rarely mentioned in semi-supervised setting. We suspect it is because previous ResNet-based models are too poor to achieve promising results on it. As we upgrade the encoder to DINOv2, we believe the semi-supervised results on ADE20K will be greatly improved. We hope it can serve as the main evaluation benchmark for future works.

COCO [128] is a large and complex dataset. It is especially popular in object detection. In our semantic segmentation task, we use its 2017 version, containing 118287 labeled images and 81 classes. There are some pioneering attempts on this dataset in SSS, including our UniMatch V1.

Following previous practices, the labeled dataset is sampled from the entire dataset as a subset, and the remaining images are treated as unlabeled images.

4.2 Implementation Details

We use the simple DPT [114] as our semantic segmentation model, built on DINOv2 [47]. We mainly report the results under DINOv2-Small and DINOv2-Base. We apply the Complementary Dropout with a probability 0.5. We adopt the same data augmentations as UniMatch V1 [1]. Specifically, weak augmentations $\mathcal{A}^w$ include random resizing between 0.5-2.0, random cropping, and horizontal flipping with probability 0.5. Strong augmentations $\mathcal{A}^s$ contain color jittering, grayscaling, gaussian blurring, and CutMix [26]. Since the patch size of DINOv2 is 16, the training resolution has to be a multiplier of 14. For Pascal, ADE20K, and COCO, we use the training (*i.e.*, cropped) size of 518, while for Cityscapes, the size is set as 798.

On Pascal and COCO, a mini-batch is evenly composed of 16 labeled images and 16 unlabeled images, while on Cityscapes and ADE20K, there are 8+8 labeled+unlabeled images. Unless otherwise specified, we conduct all experiments with four A100 GPUs.

We use the AdamW [129] optimizer with weight decay of 0.01 for training. For the most important hyper-parameter learning rate (LR), we carefully seek the optimal value for the new DINOv2 encoder. Finally, for all datasets, we set the LR of the pre-trained encoder as 5e-6, and set the LR of the randomly initialized decoder as $40\times$ larger (*i.e.*, 2e-4). We adopt a poly scheduler to decay the initial learning rate: $\text{lr} \leftarrow \text{lr} \times (1 - \frac{\text{iter}}{\text{total\_iter}})^{0.9}$. The model is trained for 60, 180, 60, and 20 epochs on Pascal, Cityscapes, ADE20K, and COCO, respectively. On Cityscapes, same as previous works, we adopt an online hard example mining (OHEM) loss [130] for labeled images, while in other cases, we adopt the standard cross-entropy loss. The confident threshold τ is set as 0.95 in all our experiments.

During inference, we only slightly interpolate the images to ensure their height and width are divisible by 14. On Cityscapes, following previous practice, we perform sliding window evaluation of the window size 798. We report the mean Intersection-over-Union (mIoU) metric (↑) of the EMA

Cityscapes SOTAs	Venue	Encoder	Ratio and absolute number of labeled images				#Params
			1/16 (186)	1/8 (372)	1/4 (744)	1/2 (1488)	
Labeled Only	-	RN-101	66.3	72.8	75.0	78.0	59.5M
U^{2}PL [119]	CVPR'22	RN-101	74.9	76.5	78.5	79.1	59.5M
PS-MT [113]	CVPR'22	RN-101	-	76.9	77.6	79.1	59.5M
GTA-Seg [120]	NeurIPS'22	RN-101	69.4	72.0	76.1	-	59.5M
PCR [121]	NeurIPS'22	RN-101	73.4	76.3	78.4	79.1	59.5M
iMAS [103]	CVPR'23	RN-101	74.3	77.4	78.1	79.3	59.5M
AugSeg [102]	CVPR'23	RN-101	75.2	77.8	79.6	80.4	59.5M
UniMatch V1 [1]	CVPR'23	RN-101	76.6	77.9	79.2	79.5	59.5M
UniMatch V1 [1]	CVPR'23	CLIP-B	76.6	78.2	79.1	79.6	88.0M
Diverse CoT [122]	ICCV'23	RN-101	75.7	77.4	78.5	-	59.5M
ESL [123]	ICCV'23	RN-101	75.1	77.2	78.9	80.5	59.5M
LogicDiag [124]	ICCV'23	RN-101	76.8	78.9	80.2	81.0	59.5M
DAW [28]	NeurIPS'23	RN-101	76.6	78.4	79.8	80.6	59.5M
DDFP [125]	CVPR'24	RN-101	77.1	78.2	79.9	80.8	59.5M
CorrMatch [12]	CVPR'24	RN-101	77.3	78.5	79.4	80.4	59.5M
AllSpark [32]	CVPR'24	MiT-B5	78.3	79.2	80.6	81.4	89.3M
BeyondPixels [29]	ECCV'24	RN-101	78.5	79.2	80.9	81.3	59.5M
SemiVL [31]	ECCV'24	CLIP-B	77.9	79.4	80.3	80.6	88.0M
Labeled Only (83.8)	-	DINOv2-S	77.2	80.2	81.7	82.4	**24.8M**
UniMatch V2	Ours		80.6	81.9	82.4	82.6	
Labeled Only (85.2)	-	DINOv2-B	80.8	82.7	84.0	84.4	97.5M
UniMatch V2	Ours		**83.6**	**84.3**	**84.5**	**85.1**	

TABLE 2: Comparison with state-of-the-art methods on **Cityscapes**. The number (*e.g.*, 83.8) next to "Labeled Only" denotes the fully-supervised result (2975 labeled images).

ADE20K SOTAs	Venue	Encoder	Ratio and absolute number of labeled images					#Params
			1/64 (316)	1/32 (631)	1/16 (1263)	1/8 (2526)	1/4 (5052)	
CutMix [23]	BMVC'20	RN-101	-	26.2	29.8	35.6	-	59.5M
AEL [100]	NeurIPS'21	RN-101	-	28.4	33.2	38.0	-	59.5M
UniMatch V1 [1]	CVPR'23	RN-101	21.6	28.1	31.5	34.6	-	59.5M
UniMatch V1 [1]	CVPR'23	CLIP-B	25.3	31.2	34.4	38.0	-	88.0M
SemiVL [31]	ECCV'24	CLIP-B	33.7	35.1	37.2	39.4	-	88.0M
Labeled Only (49.0)	-	DINOv2-S	26.1	32.7	37.1	39.8	42.7	**24.8M**
UniMatch V2	Ours		31.5	38.1	40.7	44.4	45.8	
Labeled Only (54.1)	-	DINOv2-B	32.1	39.3	42.8	46.4	49.0	97.5M
UniMatch V2	Ours		**38.7**	**45.0**	**46.7**	**49.8**	**52.0**	

TABLE 3: Comparison with state-of-the-art methods on **ADE20K**. The number (*e.g.*, 49.0) next to "Labeled Only" denotes the fully-supervised result (20210 labeled images).

teacher model. In most cases, it is approximately 0.1% - 0.2% better than the student after full convergence.

4.3 Comparison with State-of-the-Art Methods

First, we want to emphasize our comparisons with previous state-of-the-art works are unfair, due to different encoders. We use much stronger encoders than other works, which is one of the main motivations of this work. Through the unfair comparisons, we hope to clearly reveal the superiority of DINOv2 in our SSS tasks, and appeal to more works to shift to this modern encoder. We will present *fair ablation studies* under the same encoder in the next section.

Pascal: As shown in Table 1, our DINOv2-S-based UniMatch V2 framework significantly outperforms previous ResNet-101-based frameworks. For example, under the setting of 1/8 (183) labeled images, the SOTA result with ResNet-101 is 78.6% (reported in latest ECCV'24), while our result with DINOv2-S is 85.5% (+6.9%), even with over 2× fewer model parameters (59.5M *vs.* 24.8M). Such a remarkable gain has never been achieved previously by modifying the frameworks. Moreover, compared with the latest works SemiVL [31] (based on CLIP [40]) and AllSpark [32] (based on the MiT-B5 [52]), our advantages are still huge. E.g., under the 1/4 (366) setting, our DINOv2-B result surpasses AllSpark and SemiVL by 9.1% (79.8% → 88.9%) and 2.9% (86.0% → 88.9%), respectively. We also report the "Labeled only" results, where only labeled images are used for training. We can observe that the labeled-only performance of DINOv2-S is even much better than the semi-supervised results of ResNet-101 in most cases. All these comparisons clearly demonstrate the necessity of upgrading the pre-trained encoder from weak ResNet/CLIP/MiT to strong DINOv2. And the most lightweight DINOv2-S is cheaper to train than ResNet-101. Besides, we use fewer epochs than previous works (60 *vs.* 80 epochs).

Cityscapes: As exhibited in Table 2, across all evaluated splits, our results with the smallest DINOv2-S (24.8M) encoder are superior to all other frameworks based on ResNet-101 (59.5M), MiT-B5 (89.3M), or CLIP (88.0M) encoders. On the 1/16 split, our DINOv2-***S***-based UniMatch V2 improves SemiVL by 2.7% (77.9% → 80.6%), and our DINOv2-***B***-based result even outperforms it by 5.7% (77.9% → 83.6%). We

COCO SOTAs	Venue	Encoder	Ratio and absolute number of labeled images					#Params
			1/512 (232)	1/256 (463)	1/128 (925)	1/64 (1849)	1/32 (3697)	
Labeled Only	-	XC-65	22.9	28.0	33.6	37.8	42.2	54.7M
PseudoSeg [24]	ICLR'21	XC-65	29.8	37.1	39.1	41.8	43.6	54.7M
PC^2Seg [131]	ICCV'21	XC-65	29.9	37.5	40.1	43.7	46.1	54.7M
UniMatch V1 [1]	CVPR'23	XC-65	31.9	38.9	44.4	48.2	49.8	54.7M
UniMatch V1 [1]	CVPR'23	CLIP-B	36.6	44.1	49.1	53.5	55.0	88.0M
CISC-R [11]	TPAMI'23	XC-65	32.1	40.2	42.3	-	-	54.7M
LogicDiag [124]	ICCV'23	XC-65	33.1	40.3	45.4	48.8	50.5	54.7M
AllSpark [32]	CVPR'24	MiT-B5	34.1	41.7	45.5	49.6	-	89.3M
SemiVL [31]	ECCV'24	CLIP-B	**50.1**	52.8	53.6	55.4	56.5	88.0M
Labeled Only (62.5)	-	DINOv2-S	29.4	35.6	44.6	49.2	52.0	**24.8M**
UniMatch V2	Ours	DINOv2-S	39.3	45.4	53.2	55.0	57.0	**24.8M**
Labeled Only (66.4)	-	DINOv2-B	36.8	45.8	52.1	56.2	59.5	97.5M
UniMatch V2	Ours	DINOv2-B	47.9	**55.8**	**58.7**	**60.4**	**63.3**	97.5M

TABLE 4: Comparison with state-of-the-art methods on **COCO**. The number (*e.g.*, 62.5) next to "Labeled Only" denotes the fully-supervised result (118287 labeled images).

Ablation (DINOv2-B)	Pascal					ADE20K					COCO		
	1/16 (92)	1/8 (183)	1/4 (366)	1/2 (732)	Full (1464)	1/64 (316)	1/32 (631)	1/16 (1263)	1/8 (2526)	1/4 (5052)	1/512 (232)	1/256 (463)	1/128 (925)
Labeled Only	76.9	82.1	85.3	87.2	88.3	32.1	39.3	42.8	46.4	49.0	36.8	45.8	52.1
FixMatch [25]†	83.8	87.4	88.2	89.9	90.2	37.8	43.2	46.4	49.5	51.0	46.7	53.7	57.4
UniMatch V1 [1]†	86.0	87.5	88.7	**90.0**	90.4	36.7	42.8	45.9	**49.9**	51.1	45.5	54.3	57.7
UniMatch V2	**86.3**	**87.9**	**88.9**	**90.0**	**90.8**	**38.7**	**45.0**	**46.7**	49.8	**52.0**	**47.9**	**55.8**	**58.7**

TABLE 5: Ablation study against baseline methods (Labeled Only, FixMatch) and UniMatch V1 on Pascal, ADE20K and COCO. †: Different from their original implementations, we use an EMA teacher to produce stable and better pseudo labels.

also report the fully-supervised result in the bracket next to the "Labeled Only", where all available labeled images (2975 images) are used for training. Our semi-supervised result with half of the labeled images is almost equal to the fully-supervised upper bound (85.1% *vs.* 85.2%). It shows the effectiveness of our semi-supervised algorithm in substantially reducing the annotation cost. It is also worth highlighting that Pascal and Cityscapes are the two easiest benchmarks in semantic segmentation, but the performance improvement is still so tremendous, further revealing the inferiority of previously adopted encoders. We believe evaluating SSS frameworks under the modern DINOv2 encoder will attract more audiences to our field in the future.

ADE20K: Among all existing works, only SemiVL [31] reports their performance on this challenging dataset, with some reproduced results of other methods. As compared in Table 3, our DINOv2-***B***-based UniMatch V2 outperforms CLIP-***B***-based SemiVL by nearly 10% in most settings, *e.g.*, 35.1% *vs.* 45.0% (+9.9%) with 1/32 labeled images, and 39.4% *vs.* 49.8% (+10.4%) with 1/8 labeled images. Moreover, even our DINOv2-***S***-based results are much better than SemiVL on three out of four settings, achieved with 3.5× fewer model parameters (88.0M *vs.* 24.8M). On such a complex dataset, the advantages of DINOv2-based UniMatch V2 are even more significant. Moreover, we can find there is still a considerable margin between our semi-supervised results (*e.g.*, 49.8% under 1/8 splits) and the fully-supervised result (54.1%), indicating there is still much room to further improve our UniMatch V2 results.

COCO: In Table 4, compared with our UniMatch V1 [1] based on the Xception-65 [132] encoder, our lightweight V2 framework improves it remarkably, *e.g.*, 44.4% → 53.2% on the 1/128 split. Our semi-supervised algorithm also boosts the labeled-only baseline impressively, as large as +10% (45.8% → 55.8%) on the 1/256 split, highlighting the effectiveness of our framework and the value of extra unlabeled data. Furthermore, our DINOv2-B-based performance is superior to the best-performed SemiVL [31] on four of five splits, *e.g.*, 56.5% → 63.3% (+6.8%) on the 1/32 split. However, we witness the only one inferior result of our V2: on the 1/512 split, SemiVL is better than us by 2.2% (50.1% *vs.* 47.9%), indicating delicate designs may be required under such extremely label-scarce regimes.

4.4 Ablation Studies

Unless otherwise specified, we conduct all ablation studies under the DINOv2-B encoder, which is more capable than DINOv2-S and thus can provide more room for different designs to show their effectiveness. Distinguished from all existing works that only perform ablation studies on a single dataset with limited splits, we ablate our various designs and choices across extensive splits and datasets. Through these comprehensive results, we hope to shed more light on future works to better improve SSS performance.

4.4.1 *Comparison with FixMatch and UniMatch V1*

Similar to UniMatch V1 [1], our UniMatch V2 is also built on the most basic FixMatch framework [25]. Hence, we primarily present the most important comparison with the FixMatch baseline and our precedent V1 work under the same DINOv2-B encoder in Table 5. It is worth noting that, different from their original implementation that uses the online model for pseudo labeling, we re-implement them by maintaining an EMA teacher to produce higher-quality

Reference	Ablation (DINOv2-B) $\mathcal{A}^{img}$	$\mathcal{A}^{feat}$	#Streams	CompDrop	Pascal 1/16 (92)	1/8 (183)	1/4 (366)	1/2 (732)	Full (1464)	ADE20K 1/64 (316)	1/32 (631)	1/16 (1263)	1/8 (2526)	1/4 (5052)
FixMatch	✓		1	N.A.	83.8	87.4	88.2	89.9	90.2	37.8	43.2	46.4	49.5	51.0
Figure 4a	✓	✓	1	N.A.	84.7	86.6	87.7	89.6	90.2	38.1	43.6	46.0	49.8	51.3
Figure 4b	✓		2	N.A.	86.0	86.3	88.8	**90.1**	90.3	37.3	43.7	46.3	49.6	51.8
Figure 4c	✓	✓	2		82.9	86.1	88.6	89.3	90.0	37.2	**45.0**	46.4	**49.9**	51.4
UniMatch V2	✓	✓	2	✓	**86.3**	**87.9**	**88.9**	90.0	**90.8**	**38.7**	**45.0**	**46.7**	49.8	**52.0**

TABLE 6: Ablation study on various choices of learnable streams. $\mathcal{A}^{img}$ denotes image-level augmentations, while $\mathcal{A}^{feat}$ represents feature-level augmentations (*i.e.*, channel-wise Dropout). The "CompFeat" is short for complementary Dropout. Our UniMatch V2 stands out as the best design among them.

Frozen	Encoder	Pascal 1/16	1/8	1/4	1/2	Full	Cityscapes 1/16	1/8	1/4	1/2	ADE20K 1/64	1/32	1/16	1/8	1/4	COCO 1/512	1/256	1/128	1/64	1/32
		79.0	85.5	85.9	86.7	87.8	80.6	81.9	82.4	82.6	31.5	38.1	40.7	44.4	45.8	39.3	45.4	53.2	55.0	57.0
✓	DINOv2-S	81.8	83.8	84.0	85.7	86.2	75.1	77.1	78.2	78.6	29.9	34.5	37.3	39.6	41.4	37.4	42.4	48.2	51.1	52.3
		+2.8	-1.7	-1.9	-1.0	-1.6	-5.5	-4.8	-4.2	-4.0	-1.6	-3.6	-3.4	-4.8	-4.4	-1.9	-3.0	-5.0	-3.9	-4.7
		86.3	87.9	88.9	90.0	90.8	83.6	84.3	84.5	85.1	38.7	45.0	46.7	49.8	52.0	47.9	55.8	58.7	60.4	63.3
✓	DINOv2-B	84.6	87.2	87.4	88.5	90.2	79.8	81.4	81.9	82.8	35.6	40.2	42.3	45.6	47.3	43.9	50.1	54.8	57.2	58.4
		-1.7	-0.7	-1.5	-1.5	-0.6	-3.8	-2.9	-2.6	-2.3	-3.1	-4.8	-4.4	-4.2	-4.7	-4.0	-5.7	-3.9	-3.2	-4.9

TABLE 7: Ablation study on fine-tuning (by default) or freezing the pre-trained DINOv2 encoder with our UniMatch V2 framework. The frozen practice is 2× faster than fully fine-tuning in terms of training efficiency.

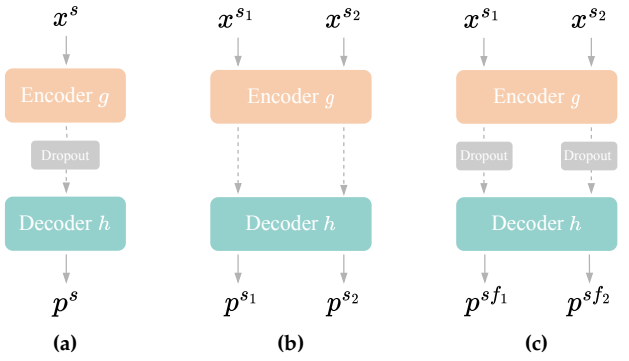


Fig. 4: Other potential designs of the learnable streams. **(a)** A single unified image-level and feature-level augmentation stream. **(b)** Two image-level augmentation streams. **(c)** Two unified image-level and feature-level augmentation streams, using random Dropout rather than our complementary Dropout. See Section 4.4.2 and Table 6 for details.

pseudo labels (same as our V2). As exhibited in Table 5, despite the strong performance of the reproduced FixMatch, our UniMatch V2 further improves it across all splits of three datasets. For instance, on ADE20K, our UniMatch outperforms FixMatch by 1.8% (43.2% → 45.0%) on the 1/32 split, and on COCO, we improve it by 2.1% (53.7% → 55.8%) on the 1/256 split. Moreover, compared with UniMatch V1, we not only speed up its training by reducing learnable streams, but also surpass it across almost all the settings (only 1 of 13 settings is 0.1% inferior), due to our carefully designed unified complementary augmentations. Notably, on the 1/32 split of ADE20K, our V2 is evidently superior to V1 by 2.2% (42.8% *vs.* 45.0%).

4.4.2 Other Variants of the Learnable Stream Design

We design two learnable streams with image-level augmentations and complementary Dropout to effectively harness unlabeled images. To convincingly demonstrate the superiority and necessity of such a design, we further explore three alternative designs on the learnable streams, as shown in Figure 4. Concretely, we attempt to 1) only use a *single* unified augmentation stream with image augmentations and a random Dropout (Figure 4a), 2) use dual-stream image augmentations, *without feature augmentations* (Figure 4b), and 3) use dual-stream unified augmentations, but adopting two *independent* Dropouts, instead of our carefully designed complementary Dropout (Figure 4c). As comprehensively validated in Table 6, the design of our UniMatch V2 is more effective than all other counterparts. Notably, on Pascal with 183 labeled images, we outperform the other three designs by 1.3%, 1.6%, and 1.8%, respectively.

4.4.3 Fine-tuning vs. Freezing the DINOv2 Encoder

Unless otherwise specified in this paper, we fully fine-tune the entire model (*i.e.*, encoder + decoder). However, limited by the GPU memory, some academic groups cannot afford such a training strategy. In view of this, we further attempt to freeze the pre-trained DINOv2 encoder and solely train the randomly initialized DPT decoder, which only accounts for 10% of the total parameters. This greatly reduces the training cost, *e.g.*, GPU memory consumption (DINOv2-B on Pascal) is reduced from 23G to 10G, and training time is reduced by 59%. The performance of fine-tuning or freezing the encoder is listed in Table 7. It is within expectation that fine-tuning achieves better results than freezing the encoder. Among all the 38 settings (spanning various datasets, splits, encoders), only on the Pascal 1/16 split with DINOv2-S, fine-tuning the encoder is inferior to freezing it, with a gap of 2.8% (79.0% *vs.* 81.8%). For the remaining 37 settings, fine-tuning clearly exceeds freezing by 0.6% - 5.5%.

Carefully analyzing the gap between fine-tuning and freezing, we can find their gap is larger on challenging datasets, such as ADE20K and COCO, with an average gap of around 3.5%, while on the easiest Pascal, the average

Scaling Encoder	#Params	Method	Pascal					ADE20K				
			1/16 (92)	1/8 (183)	1/4 (366)	1/2 (732)	Full (1464)	1/64 (316)	1/32 (631)	1/16 (1263)	1/8 (2526)	1/4 (5052)
DINOv2-Small	24.8M	Labeled Only	67.0	75.6	81.8	83.7	85.6	26.1	32.7	37.1	39.8	42.7
		UniMatch V2	79.0 (+12.0)	85.5 (+9.9)	85.9 (+4.1)	86.7 (+3.0)	87.8 (+2.2)	31.5 (+5.4)	38.1 (+5.4)	40.7 (+3.6)	44.4 (+4.6)	45.8 (+3.1)
DINOv2-Base	97.5M	Labeled Only	76.9	82.1	85.3	87.2	88.3	32.1	39.3	42.8	46.4	49.0
		UniMatch V2	86.3 (+9.4)	87.9 (+5.8)	88.9 (+3.6)	90.0 (+2.8)	90.8 (+2.5)	38.7 (+6.6)	45.0 (+5.7)	46.7 (+3.9)	49.8 (+3.4)	52.0 (+3.0)
DINOv2-Large	335.6M	Labeled Only	76.9	82.4	86.8	89.4	90.7	33.2	39.9	45.0	49.2	51.8
		UniMatch V2	86.4 (+9.5)	87.6 (+5.2)	90.3 (+3.5)	91.1 (+1.7)	91.2 (+0.5)	40.0 (+6.8)	45.0 (+5.1)	49.3 (+4.3)	52.1 (+2.9)	54.2 (+2.4)

TABLE 8: Ablation study on scaling up the encoder capacity. Updating the encoder from DINOv2-B to a larger DINOv2-L can further improve our UniMatch V2 results. Across three encoder scales, we all demonstrate the effectiveness of our proposed framework in leveraging unlabeled images.

Encoders	#Params	ADE20K			Pascal
		1/64 (316)	1/16 (1263)	1/4 (5052)	1/8 (183)
ResNet-152 [51]	78.6M	23.2	32.2	37.2	76.2
DINOv2-S [47]	24.8M	**31.5**	**40.7**	**45.8**	**85.5**
SAM-B [7]	99.8M	21.4	31.5	36.1	63.2
MiT-B5 [52]	99.2M	27.9	37.4	42.8	81.4
BEiT-B [45]	97.2M	29.4	39.6	44.2	84.5
DINOv2-B [47]	97.5M	**38.7**	**46.7**	**52.0**	**87.9**
SAM-L [7]	338.6M	30.6	39.4	44.0	77.4
BEiT-L [45]	335.9M	34.6	43.7	50.1	85.1
DINOv2-L [47]	335.6M	**40.0**	**49.3**	**54.2**	**87.6**

TABLE 9: Ablation study on the capability of various pre-trained encoders with our UniMatch V2 framework. In all cases, we cascade a DPT decoder [114] upon the encoder. We count the total parameters of the encoder and the decoder.

gap is only 1.4%. Although Cityscapes contains even fewer classes than Pascal, the gap on it is much larger, indicating that there may exist a larger distribution shift between the DINOv2 pre-training set and Cityscapes. Lastly, we want to highlight that even the frozen smallest DINOv2-S can remarkably outperform previous fine-tuned larger ResNet-101 on Pascal, ADE20K, and COCO, also with a reduction in training cost (GPU memory ↓ 47%). Motivated by the strong results achieved by a frozen encoder, we believe future works can come up with better strategies (*e.g.*, LoRA [133]) to utilize the powerful encoder.

4.4.4 Scaling Up the Capacity of the DINOv2 Encoder

Till now, we have reported many results under the DINOv2-S and DINOv2-B encoders. In practice, they are sufficient to obtain promising results. As far as we know, the largest models adopted by previous SSS works [31], [32] contain nearly 100M parameters, similar to our DINOv2-B-based DPT. Nevertheless, we are still curious whether the SSS results can be further improved by simply scaling up the capacity of the segmentation model. Therefore, we evaluate UniMatch V2 with a DINOv2-***L***-based DPT model, containing 335.6M parameters (3.5× larger than DINOv2-B). In Table 8, we reveal that our results can be further enhanced by updating the encoder from DINOv2-B to DINOv2-L. For example, on the 1/16 split of ADE20K, our result is improved from 46.7% → 49.3% (+2.6%). Additionally, even with the strongest DINOv2-L encoder, our UniMatch V2 still consistently improves the labeled-only results, showcasing its advantages in utilizing unlabeled images.

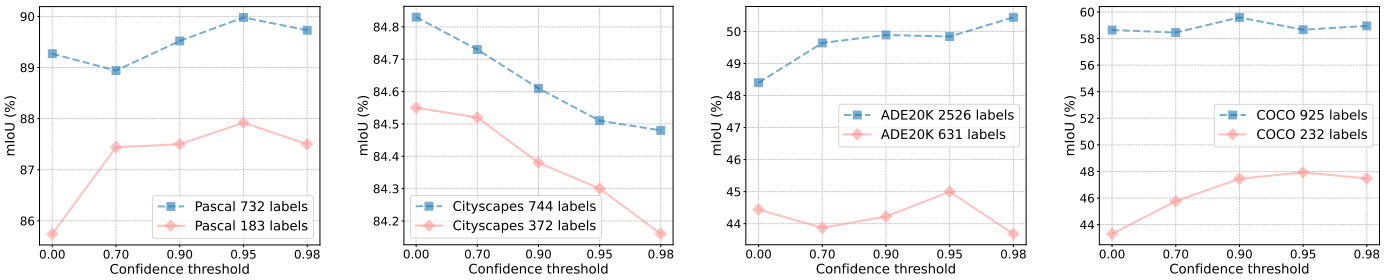
4.4.5 Comparison with Other Pre-trained Encoders

We here compare our adopted DINOv2 with other capable pre-trained vision encoders, including ResNet-152 [51], MiT-B5 [52], BEiT-B [45], BEiT-L [45], SAM-B [7], and SAM-L [7]. It is worth noting that we carefully re-find an optimal learning rate for these encoders, instead of directly using our DINOv2's learning rate. We apply the same UniMatch V2 framework for these encoders, except BEiT. When adopting BEiT, we use the student model to produce pseudo labels, because we find its EMA teacher is very poor, probably due to the relative positional encoding.

As shown in Table 9, we divide all encoders into three groups based on their number of parameters. There are four key observations. (1) Within each group, our adopted DINOv2 encoder significantly outperforms other alternatives. Our DINOv2-S is better than ResNet-152 by 8% on average, also requiring 3× fewer parameters. Compared with the strong BEiT-L [45] encoder on ADE20K, our DINOv2-L results are 5% higher on average of three splits. (2) Our results under smaller DINOv2 are even better than other larger models. DINOv2-S consistently surpasses BEiT-B, and DINOv2-B is superior to BEiT-L. (3) Although SAM [7] is specifically designed for image segmentation, its encoder performs poorly in our SSS task, mainly due to its weak semantic capability acquired by the class-agnostic segmentation pre-training task. (4) Some prior works fail to fully unlock the capability of advanced encoders. E.g., the results of AllSpark [32] (Table 1) are much inferior to our UniMatch V2 under the same MiT-B5 encoder (Table 9), *e.g.*, 78.4% *vs.* 81.4% on the Pascal 183 split.

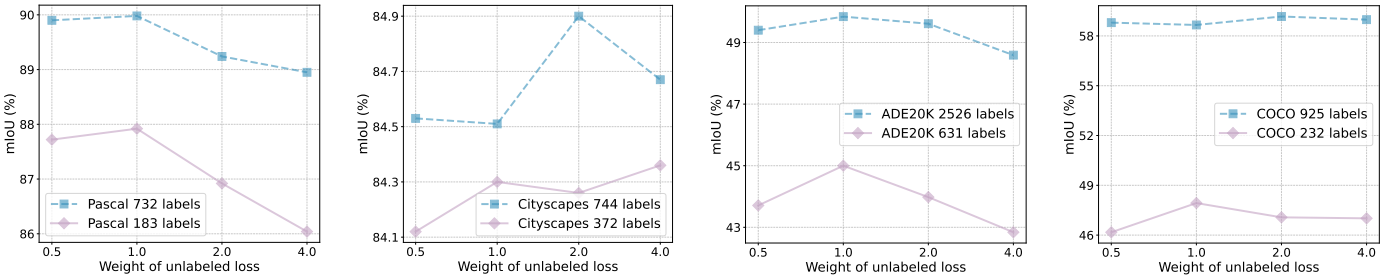
4.4.6 Values of the Confidence Threshold

As aforementioned, we follow FixMatch [25] to use a confidence threshold τ to discard low-confident noisy pseudo labels. By default, we set it as 0.95 for all settings. Here, we ablate its different values across all the four evaluated datasets with two labeled splits each. As exhibited in Figure 5, we try five values of τ: 0 (not filtering), 0.7, 0.90, 0.95 (default), and 0.98. It can be concluded that in many cases (4 out of 8 settings), 0.95 is already the optimal value for τ and



(a) Pascal **(b)** Cityscapes **(c)** ADE20K **(d)** COCO

Fig. 5: Ablation study on the confidence threshold τ (0.95 by default) used to select high-quality pseudo labels.

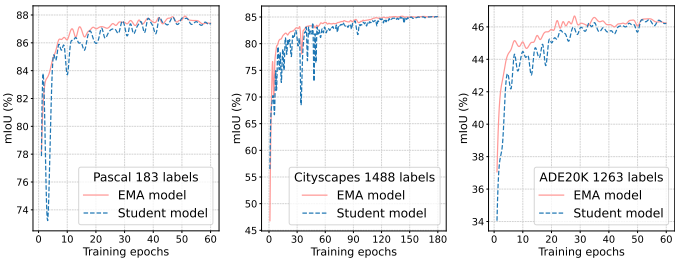


(a) Pascal **(b)** Cityscapes **(c)** ADE20K **(d)** COCO

Fig. 6: Ablation study on the weight λ of unlabeled loss $\mathcal{L}^u$. The default unlabeled loss weight 1.0 yields the best results.

EMA Teacher	Pascal		ADE20K		COCO		
	1/16 (92)	1/8 (183)	1/64 (316)	1/32 (631)	1/512 (232)	1/256 (463)	1/128 (925)
✓	**86.3**	**87.9**	**38.7**	**45.0**	**47.9**	**55.8**	**58.7**
	83.5	86.5	37.7	42.8	44.9	54.1	57.6

TABLE 10: Ablation study on whether to use an EMA teacher or directly use the online student to produce pseudo label.



(a) Pascal **(b)** Cityscapes **(c)** ADE20K

Fig. 7: Comparison between the performance of the online trained student model and the EMA teacher model.

yields the most consistent performance. A clear exception is the Cityscapes dataset (Figure 5b), where there is a negative correlation between the value of τ and model performance: the larger τ is, the lower performance is. The optimal τ for Cityscapes is 0, which means all pseudo labels are used for training without filtering. This is indeed the same value set by UniMatch V1 for Cityscapes. However, in this V2 work, we avoid specifically fine-tuning this value for each dataset and maintain the same τ for all datasets.

4.4.7 *Values of the Weight λ of Unlabeled Loss*

By default, we set the weight λ of unlabeled loss $\mathcal{L}^u$ as 1.0, the same as that of labeled loss. Here we ablate other values (*i.e.*, 0.5, 2.0, 4.0) of the unlabeled loss weight. As shown in Figure 6, the default weight 1.0 is almost the optimal value, better than all other weights in 6 out of 8 settings (spanning all the four datasets, each with two labeled data splits). In most cases, when the loss weight has exceeded 1.0, the larger the value is, the worse performance will be.

4.4.8 *Effectiveness of the EMA Teacher*

Different from FixMatch and our previous UniMatch V1, we use an EMA teacher to produce pseudo labels for the online student to learn in V2. This modification is motivated by observations that the EMA model is mostly 1% - 2% better than the online model at early training stages and its performance is more stable, as visualized in Figure 7. In Table 10, we further compare the final performance of using the EMA teacher *vs.* using the online student to produce pseudo labels. In the former case (our default practice), we report the EMA teacher test results. In the latter case, we report better test results of the online model and its EMA version. For all evaluated settings of three datasets, using the EMA model yields much better results, further proving its higher labeling quality.

4.4.9 *Dual-Stream Learning vs. Doubled Batch Size*

With the design of dual complementary streams, our UniMatch V2 is significantly superior to FixMatch. However, this also comes with nearly doubled training cost. Therefore, we need examine whether FixMatch can be on par with ours if training it with doubled batch size for doubled epochs, *i.e.*, whether our better results can be trivially achieved by simply increasing the training budget. As shown in Table 11, although we manually increase the training cost of FixMatch by two times, it is still evidently inferior to our UniMatch

Method	Pascal		ADE20K		COCO
	1/16	1/8	1/64	1/32	1/512
FixMatch + 2× BS & Epoch	84.4	86.7	36.8	44.4	47.5
Our UniMatch V2	**86.3**	**87.9**	**38.7**	**45.0**	**47.9**

TABLE 11: Our UniMatch V2, designed with complementary dual streams, is superior to the FixMatch naively augmented by doubled batch size and doubled epochs.

Dropout Position	Pascal				ADE20K		COCO
	1/16	1/8	1/4	1/2	1/32	1/16	1/256
Encoder - Decoder	**86.3**	**87.9**	**88.9**	**90.0**	**45.0**	46.7	**55.8**
Decoder - Classifier	83.5	87.1	**88.9**	89.7	43.5	**46.9**	54.6

TABLE 12: Ablation study on different inserted positions of our complementary Dropout.

V2 on the three datasets, showcasing the effectiveness of our designed complementary views.

4.4.10 *Inserted Position of the Complementary Dropout*

Following UniMatch V1, by default, we insert our proposed complementary Dropout at the intersection of the encoder (*e.g.*, DINOv2) and the decoder (*e.g.*, DPT head). We also tried to move this mechanism to the intersection of the decoder and the final convolutional classifier. Although it is inserted at the back of the model, the gradient will be backpropagated throughout the entire model, enhancing the overall robustness. As compared in Table 12, between the two attempted positions, the encoder-decoder position is generally better than the decoder-classifier position, same as the observation in UniMatch V1. It demonstrates that it is more beneficial to inject feature-level augmentations in the middle of the model.

4.5 Towards A More Real-World SSS Setting

We hope to highlight that, existing SSS works *all* adopt an *artificial* semi-supervised setting, where labeled images are selected from all available labeled data as a small subset, and manually treat the remaining ones as unlabeled images, *i.e.*, not use their ground-truth labels.

However, a more real-world and practical SSS setting should be: we directly use *all available* labeled images as our labeled set, and seek additional *truly unlabeled* images as our unlabeled set. This setting is much more challenging than existing settings, since the competitive fully-supervised baseline is hard to further boost. But we believe it is necessary to make efforts towards this, since the ultimate goal of SSS is to benefit real-world applications. In most real-world scenarios, substantial images (*e.g.*, 10K) have been annotated over time, but meantime a much larger pool of images (*e.g.*, 100K) have not yet been manually labeled.

To this end, we evaluate our UniMatch V2 under three challenging but practical settings: (1) we use COCO all 118K labeled images and its officially provided 123K unlabeled images, (2) we use ADE20K all 20K labeled images, and treat the COCO 118K labeled images as our unlabeled data, and (3) we use Cityscapes all 3K labeled images, and treat its additionally provided 20K images without precise labels as our unlabeled pool. All the results are summarized in Table 13.

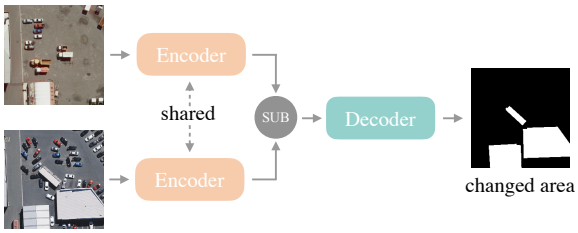


Fig. 8: A widely adopted pipeline for remote sensing change detection. Features extracted from bi-temporal images are subtracted and then sent into the decoder for binary segmentation.

Labeled Data (# Img)	+ Unlabeled Data (# Img)	Improvement
COCO (118K)	COCO Extra (123K)	66.4 → 67.1
ADE20K (20K)	COCO Labeled (118K)	54.1 → 54.9
ADE20K (20K)	COCO All (118K + 123K)	**54.1** → **55.7**
Cityscapes (3K)	Cityscapes Extra (20K)	85.2 → 85.5

TABLE 13: Making efforts towards a *real-world* and *large-scale* semi-supervised semantic segmentation setting, where many images (*e.g.*, 10K) have been labeled over time, and meantime larger-scale unlabeled images (*e.g.*, 100K) are available.

There are four observations. (1) Even already given adequate labeled images, our UniMatch V2 can still enhance the model performance by incorporating additional unlabeled images. All the three challenging fully-supervised baselines are boosted by our UniMatch V2. (2) Even if the labeled and unlabeled images come from slightly different data distributions, unlabeled images are still highly beneficial. For instance, COCO 118K unlabeled images improve the ADE20K fully-supervised result (with 20K labeled images) from 54.1% to 54.9%. (3) There is a scaling law in unlabeled data. Notably, for the ADE20K dataset, when increasing the number of unlabeled COCO images from 118K to 241K, the performance is further improved from 54.9% to 55.7%. (4) Compared with small-scale SSS settings, our improvements in such real-world large-scale scenarios appear marginal, mainly due to the high labeled-only results. But we believe such a challenging direction is meaningful to explore and we expect future works can have significant advances.

4.6 Extending UniMatch V2 to Broader Scenarios

4.6.1 *Remote Sensing Change Detection*

Remote sensing images or satellite images are extremely laborious to annotate due to ultra-high resolution. Therefore, we further investigate the effectiveness of our UniMatch V2 in the task of remote sensing change detection, which aims to spot the changed regions between two bi-temporal images. A widely used pipeline is illustrated in Figure 8. Extracted features of the dual images are subtracted and then sent into the decoder. It can be considered a binary segmentation task, but with dual inputs. In this task, we use the student itself to produce pseudo labels, because it evolves much quicker than its EMA version. We train the model for 60 epochs, with batch size 8+8 and image size 252. Other hyper-parameters (*e.g.*, learning rate, optimizer) are the same as those for natural images.

As shown in Table 14, on the two most representative change detection datasets LEVIR-CD [137] and WHU-CD [138], our UniMatch V2, even building on the lightest

Change Detection	Model	LEVIR-CD								WHU-CD			
		5% (356)		10% (712)		20% (1424)		40% (2848)		20% (1189)		40% (2378)	
		IoU^c	OA	IoU^c	OA	IoU^c	OA	IoU^c	OA	IoU^c	OA	IoU^c	OA
AdvNet [134]	FCN-RN-50	66.1	98.08	72.3	98.45	74.6	98.58	75.0	98.60	73.8	98.80	76.6	98.94
S4GAN [9]	FCN-RN-50	64.0	97.89	67.0	98.11	73.4	98.51	75.4	98.62	70.8	98.60	76.4	98.96
SemiCDNet [135]	UNet++	67.6	98.17	71.5	98.42	74.3	98.58	75.5	98.63	66.7	98.28	75.9	98.93
SemiCD [18]	FCN-RN-50	72.5	98.47	75.5	98.63	76.2	98.68	77.2	98.72	74.8	98.84	77.2	98.96
UniMatch V1 [1]	PSPNet-RN-50	75.6	98.62	79.0	98.83	79.0	98.84	78.2	98.79	82.9	99.26	84.4	99.32
UniMatch V1 [1]	DeepLab-RN-50	80.7	98.95	82.0	99.02	81.7	99.02	82.1	99.03	81.7	99.18	85.1	99.35
SemiCD-VL [136]	RN-50 + VLM	81.9	99.02	82.6	99.06	82.7	99.05	83.0	99.07	84.8	99.36	85.7	99.39
Labeled Only	DPT-DINOv2-S	78.1	98.77	80.4	98.91	81.1	98.95	82.4	99.03	79.5	99.09	82.6	99.25
UniMatch V2	DPT-DINOv2-S	82.1	99.01	82.7	99.05	83.2	99.08	83.5	99.10	86.1	99.42	87.6	99.48
Labeled Only	DPT-DINOv2-B	79.5	98.86	81.3	98.96	82.4	99.02	83.6	99.10	83.3	99.27	87.0	99.45
UniMatch V2	DPT-DINOv2-B	**83.3**	**99.08**	**83.8**	**99.11**	**84.3**	**99.14**	**84.3**	**99.14**	**87.9**	**99.50**	**88.6**	**99.52**

TABLE 14: Performance of UniMatch V2 in the scenario of remote sensing change detection. We evaluate it on two representative datasets **LEVIR-CD** [137] and **WHU-CD** [138]. The IoU^c denotes *changed-class IoU*, and OA denotes *overall accuracy*.

CIFAR-10 Classification	Seed					Mean
	0	1	2	3	4	
SimMatch [78]	95.58	95.50	95.34	94.06	95.26	95.15
ShrinkMatch [79]	95.39	95.44	95.36	94.76	95.35	95.26
UniMatch V2	95.70	95.42	95.54	95.09	95.34	**95.42**

TABLE 15: Results of our UniMatch V2 of the semi-supervised **classification** task on **CIFAR-10** with 25 labels per class.

STL-10 Classification	Seed			Mean
	0	1	2	
FixMatch [25]	90.91	88.71	90.94	90.19
FlexMatch [76]	91.35	92.29	91.66	91.77
ShrinkMatch [79]	91.13	92.43	91.10	91.55
UniMatch V2	91.70	92.91	92.75	**92.45**

TABLE 16: Results of our UniMatch V2 of the semi-supervised **classification** task on **STL-10** with 25 labels per class.

DINOv2-S encoder, consistently surpasses all prior works across all settings. When adopting the larger DINOv2-B encoder, the results are further enhanced. For instance, on the WHU-CD dataset with 20% labeled images, our DINOv2-S IoU result is superior to the previous best SemiCD-VL [136] by 1.3% (84.8% *vs.* 86.1%), while our DINOv2-B result is better than it by 3.1% (84.8% *vs.* 87.9%). These impressive results demonstrate the strong universality of our framework, even for bi-temporal SSS scenarios.

4.6.2 Semi-Supervised Classification

We further extend our UniMatch V2 framework to the more fundamental scenario of semi-supervised classification. We follow the protocols of previous works [25], [78] and conduct experiments on four representative datasets. On CIFAR-10 [140] (Table 15), STL-10 [141] (Table 16), and SVHN [142] (Table 17), we adopt FixMatch [25] with distribution alignment [67] as our baseline. For stability of reported performance, we run each labeled split with three or five different random seeds. On ImageNet-1K [50] (Table 18), we use SimMatch [78] as our baseline. As reported in the tables, on all the four datasets, our UniMatch V2 outperforms previous methods non-trivially.

SVHN	Seed (4 labels)			Mean	Seed (25 labels)			Mean
	0	1	2		0	1	2	
FixMatch [25]	94.5	96.9	97.1	96.2	98.0	98.0	97.9	98.0
FlexMatch [76]	89.2	89.9	96.3	91.8	91.8	91.8	96.7	93.4
ShrinkMatch [79]	98.0	97.8	96.7	97.5	98.1	98.1	98.0	98.0
UniMatch V2	98.2	98.2	98.1	**98.2**	98.2	98.2	98.2	**98.2**

TABLE 17: Results of our UniMatch V2 of the semi-supervised **classification** task on **SVHN** with 4 or 25 labels per class.

ImageNet-1K	CoMatch [139]	SimMatch [78]	UniMatch V2
Top-1 / Top-5 Acc	73.6 / 91.6	74.1 / 91.5	**74.3** / **91.7**

TABLE 18: Results of our UniMatch V2 of the semi-supervised **classification** task on **ImageNet-1K** with 10% labels.

4.7 Qualitative Comparisons

We qualitatively compare our current UniMatch V2 framework with our previous UniMatch V1 in Figure 9. On the four representative benchmarks, our UniMatch V2 produces sharper predictions than V1, also with less confusion to challenging semantics. Additionally, we provide qualitative comparisons with previous SOTA methods AllSpark [32] and SemiVL [31] in Figure 10. Our UniMatch V2 stands out as the most accurate semantic segmentor.

5 Conclusion

In this work, we present UniMatch V2 to strengthen our prior V1 framework for semi-supervised semantic segmentation. We update previous outdated ResNet encoders to the most capable DINOv2 encoders. We conduct comprehensive experiments for future works to compare with us in this new benchmark easily. Technically, we unify the image-level and feature-level augmentations into a single stream, and further design a Complementary Dropout to take full advantage of the dual-stream practice by crafting better dual learnable views. Consequently, our UniMatch V2 significantly outperforms all precedent works.

References

[1] L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi, "Revisiting weak-to-strong consistency in semi-supervised semantic segmentation," in *CVPR*, 2023. 1, 2, 3, 4, 5, 8, 9, 10, 15, 16

Fig. 9: Qualitative comparisons between UniMatch V1 [1] and current V2. From top to bottom, the images are sampled from Pascal, Cityscapes, ADE20K, and COCO datasets, respectively.

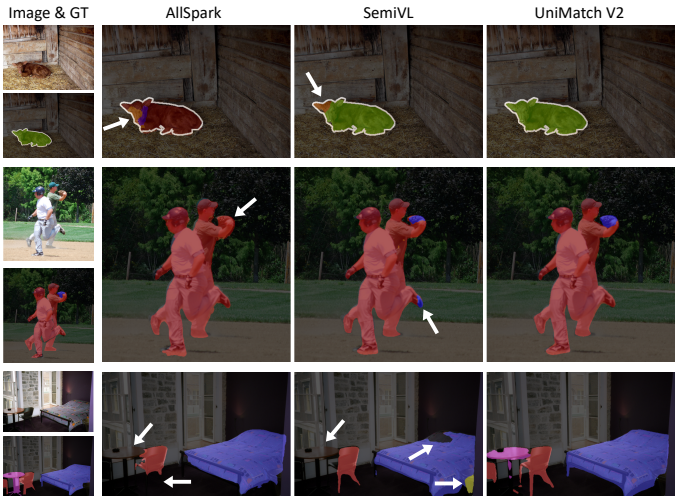


Fig. 10: Qualitative comparisons with previous state-of-the-art methods AllSpark [32] and SemiVL [31].

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015. 1

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, 2017. 1, 5

[4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017. 1

[5] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *NeurIPS*, 2021. 1

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016. 1, 7, 8

[7] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *ICCV*, 2023. 1, 2, 3, 12

[8] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *CVPR*, 2024. 1, 2, 3

[9] S. Mittal, M. Tatarchenko, and T. Brox, "Semi-supervised semantic segmentation with high-and low-level consistency," *TPAMI*, 2019. 1, 3, 15

[10] X. Chen, Y. Yuan, G. Zeng, and J. Wang, "Semi-supervised semantic segmentation with cross pseudo supervision," in *CVPR*, 2021. 1, 2, 3

[11] L. Wu, L. Fang, X. He, M. He, J. Ma, and Z. Zhong, "Querying labeled for unlabeled: Cross-image semantic consistency guided semi-supervised semantic segmentation," *TPAMI*, 2023. 1, 3, 10

[12] B. Sun, Y. Yang, L. Zhang, M.-M. Cheng, and Q. Hou, "Corrmatch: Label propagation via correlation matching for semi-supervised semantic segmentation," in *CVPR*, 2024. 1, 2, 3, 4, 8, 9

[13] L. Yu, S. Wang, X. Li, C.-W. Fu, and P.-A. Heng, "Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation," in *MICCAI*, 2019. 1, 3

[14] Y. Bai, D. Chen, Q. Li, W. Shen, and Y. Wang, "Bidirectional copy-paste for semi-supervised medical image segmentation," in *CVPR*, 2023. 1, 3

[15] Z. Zhao, Z. Wang, L. Wang, Y. Yuan, and L. Zhou, "Alternate diverse teaching for semi-supervised medical image segmentation," in *ECCV*, 2024. 1, 3

[16] H. Chi, J. Pang, B. Zhang, and W. Liu, "Adaptive bidirectional displacement for semi-supervised medical image segmentation," in *CVPR*, 2024. 1, 3

[17] J.-X. Wang, S.-B. Chen, C. H. Ding, J. Tang, and B. Luo, "Ranpaste: Paste consistency and pseudo label for semisupervised remote sensing image semantic segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, 2021. 1, 3

[18] W. G. C. Bandara and V. M. Patel, "Revisiting consistency regularization for semi-supervised change detection in remote sensing images," *arXiv:2204.08454*, 2022. 1, 3, 15

[19] S. Yuan, R. Zhong, C. Yang, Q. Li, and Y. Dong, "Dynamically updated semi-supervised change detection network combining cross-supervision and screening algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 1, 3

[20] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "St++: Make self-training work better for semi-supervised semantic segmentation," in *CVPR*, 2022. 1, 2, 3, 8

[21] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *ICML Workshop*, 2013. 1, 2, 3

[22] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020. 1, 3

[23] G. French, T. Aila, S. Laine, M. Mackiewicz, and G. Finlayson, "Semi-supervised semantic segmentation needs strong, high-dimensional perturbations," in *BMVC*, 2020. 2, 3, 9

[24] Y. Zou, Z. Zhang, H. Zhang, C.-L. Li, X. Bian, J.-B. Huang, and T. Pfister, "Pseudoseg: Designing pseudo labels for semantic segmentation," in *ICLR*, 2021. 2, 3, 10

[25] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020. 2, 3, 4, 5, 10, 12, 15

[26] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019. 2, 3, 4, 8

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, 2014. 2, 5

[28] R. Sun, H. Mai, T. Zhang, and F. Wu, "Daw: exploring the better weighting function for semi-supervised semantic segmentation," in *NeurIPS*, 2023. 2, 6, 8, 9

[29] P. Howlader, S. Das, H. Le, and D. Samaras, "Beyond pixels: Semi-supervised semantic segmentation with a multi-scale patch-based multi-label classifier," in *ECCV*, 2024. 2, 4, 6, 8, 9

[30] W. Shin, H. J. Park, J. S. Kim, and S. W. Han, "Revisiting and maximizing temporal knowledge in semi-supervised semantic segmentation," *arXiv:2405.20610*, 2024. 2, 6

[31] L. Hoyer, D. J. Tan, M. F. Naeem, L. Van Gool, and F. Tombari, "Semivl: Semi-supervised semantic segmentation with vision-language guidance," in *ECCV*, 2024. 2, 4, 6, 8, 9, 10, 12, 15, 16

[32] H. Wang, Q. Zhang, Y. Li, and X. Li, "Allspark: Reborn labeled features from unlabeled in transformer for semi-supervised semantic segmentation," in *CVPR*, 2024. 2, 4, 8, 9, 10, 12, 15, 16

[33] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv:2408.00714*, 2024. 2

[34] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv:2406.09414*, 2024. 2

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021. 2

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021. 2

[37] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. W. Lau, "Biformer: Vision transformer with bi-level routing attention," in *CVPR*, 2023. 2

[38] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *CVPR*, 2022. 2

[39] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *CVPR*, 2021. 2

[40] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021. 2, 4, 9

[41] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "ibot: Image bert pre-training with online tokenizer," in *ICLR*, 2022. 2

[42] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," in *CVPR*, 2023. 2

[43] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020. 2, 6

[44] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020. 2, 6

[45] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," in *ICLR*, 2022. 2, 3, 12

[46] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022. 2

[47] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *TMLR*, 2023. 2, 3, 4, 5, 8, 12

[48] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv:2303.15389*, 2023. 2

[49] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," in *CVPR*, 2023. 2

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009. 2, 15

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 2, 3, 5, 12

[52] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *NeurIPS*, 2021. 2, 3, 9, 12

[53] X. J. Zhu, "Semi-supervised learning literature survey," 2005. 3

[54] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, 2009. 3

[55] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, 2020. 3

[56] J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han, "Large language models can self-improve," in *EMNLP*, 2023. 3

[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012. 3

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 3

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017. 3

[60] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *NeurIPS*, 2005. 3

[61] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *WACV/MOTION*, 2005. 3

[62] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," in *NeurIPS*, 2020. 3

[63] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *CVPR*, 2021. 3

[64] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017. 3, 7

[65] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," in *NeurIPS*, 2020. 3, 4

[66] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *NeurIPS*, 2019. 3, 4

[67] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *ICLR*, 2020. 3, 4, 5, 15

[68] Z. Zhao, L. Zhou, L. Wang, Y. Shi, and Y. Gao, "Lassl: Label-guided self-training for semi-supervised learning," in *AAAI*, 2022. 3

[69] Y. Wang, H. Chen, Q. Heng, W. Hou, M. Savvides, T. Shinozaki, B. Raj, Z. Wu, and J. Wang, "Freematch: Self-adaptive thresholding for semi-supervised learning," in *ICLR*, 2023. 3

[70] R. Du, D. Chang, Z. Ma, K. Liang, Y.-Z. Song, and J. Guo, "Semi-supervised learning for fgvc with out-of-category data," *TPAMI*, 2023. 3

[71] Z. Tan, K. Zheng, and W. Huang, "Otmatch: Improving semi-supervised learning with optimal transport," in *ICML*, 2024. 3

[72] M. Li, R. Wu, H. Liu, J. Yu, X. Yang, B. Han, and T. Liu, "Instant: Semi-supervised learning with instance-dependent thresholds," in *NeurIPS*, 2023. 3

[73] Z. Li, L. Qi, Y. Shi, and Y. Gao, "Iomatch: Simplifying open-set semi-supervised learning with joint inliers and outliers utilization," in *ICCV*, 2023. 3

[74] J. Liang, R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Generalized semi-supervised learning via self-supervised feature adaptation," in *NeurIPS*, 2023. 3

[75] K. Gan and T. Wei, "Erasing the bias: Fine-tuning foundation models for semi-supervised learning," in *ICML*, 2024. 3

[76] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *NeurIPS*, 2021. 3, 15

[77] B. Chen, J. Jiang, X. Wang, P. Wan, J. Wang, and M. Long, "Debiased self-training for semi-supervised learning," in *NeurIPS*, 2022. 3

[78] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "Simmatch: Semi-supervised learning with similarity matching," in *CVPR*, 2022. 3, 15

[79] L. Yang, Z. Zhao, L. Qi, Y. Qiao, Y. Shi, and H. Zhao, "Shrinking class space for enhanced certainty in semi-supervised learning," in *ICCV*, 2023. 3, 15

[80] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *ICCV*, 2017. 3

[81] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *NeurIPS*, 2014. 3

[82] Y. Ouali, C. Hudelot, and M. Tami, "Semi-supervised semantic segmentation with cross-consistency training," in *CVPR*, 2020. 3

[83] Z. Feng, Q. Zhou, Q. Gu, X. Tan, G. Cheng, X. Lu, J. Shi, and L. Ma, "Dmt: Dynamic mutual training for semi-supervised learning," *PR*, 2022. 3

[84] Z. Ke, D. Qiu, K. Li, Q. Yan, and R. W. Lau, "Guided collaborative training for pixel-wise semi-supervised learning," in *ECCV*, 2020. 3

[85] R. Mendel, L. A. de Souza, D. Rauber, J. P. Papa, and C. Palm, "Semi-supervised segmentation based on error-correcting supervision," in *ECCV*, 2020. 3

[86] Y. Zhou, H. Xu, W. Zhang, B. Gao, and P.-A. Heng, "C3-semiseg: Contrastive semi-supervised segmentation via cross-set learning and dynamic class-balancing," in *ICCV*, 2021. 3

[87] Y. Zhong, B. Yuan, H. Wu, Z. Yuan, J. Peng, and Y.-X. Wang, "Pixel contrastive-consistent semi-supervised semantic segmentation," in *ICCV*, 2021. 3

[88] I. Alonso, A. Sabater, D. Ferstl, L. Montesano, and A. C. Murillo, "Semi-supervised semantic segmentation with pixel-level con-

trastive learning from a class-wise memory bank," in *ICCV*, 2021. 3
[89] P. Zhang, B. Zhang, T. Zhang, D. Chen, and F. Wen, "Robust mutual learning for semi-supervised semantic segmentation," *arXiv:2106.00609*, 2021. 3
[90] X. Lai, Z. Tian, L. Jiang, S. Liu, H. Zhao, L. Wang, and J. Jia, "Semi-supervised semantic segmentation with directional context-aware consistency," in *CVPR*, 2021. 3
[91] R. He, J. Yang, and X. Qi, "Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation," in *ICCV*, 2021. 3
[92] D. Guan, J. Huang, A. Xiao, and S. Lu, "Unbiased subclass regularization for semi-supervised semantic segmentation," in *CVPR*, 2022. 3
[93] S. Liu, S. Zhi, E. Johns, and A. J. Davison, "Bootstrapping semantic segmentation with regional contrast," in *ICLR*, 2022. 3
[94] D. Kwon and S. Kwak, "Semi-supervised semantic segmentation with error localization network," in *CVPR*, 2022. 3
[95] J. Zhang, T. Wu, C. Ding, H. Zhao, and G. Guo, "Region-level contrastive and consistency learning for semi-supervised semantic segmentation," in *IJCAI*, 2022. 3
[96] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv:1708.04552*, 2017. 3
[97] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998. 3
[98] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *CVPR*, 2018. 3
[99] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *ECCV*, 2018. 3
[100] H. Hu, F. Wei, H. Hu, Q. Ye, J. Cui, and L. Wang, "Semi-supervised semantic segmentation via adaptive equalization learning," in *NeurIPS*, 2021. 3, 9
[101] J. Yuan, Y. Liu, C. Shen, Z. Wang, and H. Li, "A simple baseline for semi-supervised semantic segmentation with strong data augmentation," in *ICCV*, 2021. 3
[102] Z. Zhao, L. Yang, S. Long, J. Pi, L. Zhou, and J. Wang, "Augmentation matters: A simple-yet-effective approach to semi-supervised semantic segmentation," in *CVPR*, 2023. 3, 4, 7, 8, 9
[103] Z. Zhao, S. Long, J. Pi, J. Wang, and L. Zhou, "Instance-specific and model-adaptive supervision for semi-supervised semantic segmentation," in *CVPR*, 2023. 3, 4, 8, 9
[104] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *NeurIPS*, 2020. 4
[105] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, "Unbiased teacher for semi-supervised object detection," in *ICLR*, 2021. 5
[106] Y. Tang, W. Chen, Y. Luo, and Y. Zhang, "Humble teachers teach better students for semi-supervised object detection," in *CVPR*, 2021. 5
[107] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, "End-to-end semi-supervised object detection with soft teacher," in *ICCV*, 2021. 5
[108] L. Melas-Kyriazi and A. K. Manrai, "Pixmatch: Unsupervised domain adaptation via pixelwise consistency training," in *CVPR*, 2021. 5
[109] J. Xiao, L. Jing, L. Zhang, J. He, Q. She, Z. Zhou, A. Yuille, and Y. Li, "Learning from temporal gradient for semi-supervised action recognition," in *CVPR*, 2022. 5
[110] Y. Xu, F. Wei, X. Sun, C. Yang, Y. Shen, B. Dai, B. Zhou, and S. Lin, "Cross-model pseudo-labeling for semi-supervised action recognition," in *CVPR*, 2022. 5
[111] S. Vaze, A. Vedaldi, and A. Zisserman, "No representation rules them all in category discovery," in *NeurIPS*, 2023. 5
[112] C.-W. Kuo, C.-Y. Ma, J.-B. Huang, and Z. Kira, "Featmatch: Feature-based augmentation for semi-supervised learning," in *ECCV*, 2020. 5
[113] Y. Liu, Y. Tian, Y. Chen, F. Liu, V. Belagiannis, and G. Carneiro, "Perturbed and strict mean teachers for semi-supervised semantic segmentation," in *CVPR*, 2022. 5, 7, 8, 9
[114] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *ICCV*, 2021. 5, 8, 12
[115] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *TPAMI*, 2018. 5
[116] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *NeurIPS*, 2020. 5
[117] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021. 6
[118] S. Liu, S. Zhi, E. Johns, and A. J. Davison, "Bootstrapping semantic segmentation with regional contrast," in *ICLR*, 2022. 6
[119] Y. Wang, H. Wang, Y. Shen, J. Fei, W. Li, G. Jin, L. Wu, R. Zhao, and X. Le, "Semi-supervised semantic segmentation using unreliable pseudo-labels," in *CVPR*, 2022. 6, 8, 9
[120] Y. Jin, J. Wang, and D. Lin, "Semi-supervised semantic segmentation via gentle teaching assistant," in *NeurIPS*, 2022. 8, 9
[121] H.-M. Xu, L. Liu, Q. Bian, and Z. Yang, "Semi-supervised semantic segmentation with prototype-based consistency regularization," in *NeurIPS*, 2022. 8, 9
[122] Y. Li, X. Wang, L. Yang, L. Feng, W. Zhang, and Y. Gao, "Diverse cotraining makes strong semi-supervised segmentor," in *ICCV*, 2023. 8, 9
[123] J. Ma, C. Wang, Y. Liu, L. Lin, and G. Li, "Enhanced soft label for semi-supervised semantic segmentation," in *ICCV*, 2023. 8, 9
[124] C. Liang, W. Wang, J. Miao, and Y. Yang, "Logic-induced diagnostic reasoning for semi-supervised semantic segmentation," in *ICCV*, 2023. 8, 9, 10
[125] X. Wang, H. Bai, L. Yu, Y. Zhao, and J. Xiao, "Towards the uncharted: Density-descending feature perturbation for semi-supervised semantic segmentation," in *CVPR*, 2024. 8, 9
[126] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *IJCV*, 2015. 7, 8
[127] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017. 7, 8
[128] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014. 7, 8
[129] I. Loshchilov, "Decoupled weight decay regularization," in *ICLR*, 2019. 8
[130] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016. 8
[131] Y. Zhong, B. Yuan, H. Wu, Z. Yuan, J. Peng, and Y.-X. Wang, "Pixel contrastive-consistent semi-supervised semantic segmentation," in *ICCV*, 2021. 10
[132] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017. 10
[133] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2022. 12
[134] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *CVPR*, 2019. 15
[135] D. Peng, L. Bruzzone, Y. Zhang, H. Guan, H. Ding, and X. Huang, "Semicdnet: A semisupervised convolutional neural network for change detection in high resolution remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2020. 15
[136] K. Li, X. Cao, Y. Deng, and D. Meng, "Diffmatch: Visual-language guidance makes better semi-supervised change detector," *arXiv:2405.04788*, 2024. 15
[137] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sensing*, 2020. 14, 15
[138] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Transactions on Geoscience and Remote Sensing*, 2018. 14, 15
[139] J. Li, C. Xiong, and S. C. Hoi, "Comatch: Semi-supervised learning with contrastive graph regularization," in *ICCV*, 2021. 15
[140] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009. 15
[141] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011. 15
[142] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop*, 2011. 15