Semi-Supervised Video Semantic Segmentation with Inter-Frame Feature Reconstruction

Jiafan Zhuang, Zilei Wang*, Yuan Gao
University of Science and Technology of China
{jfzhuang, gyy}@mail.ustc.edu.cn, zlwang@ustc.edu.cn

Abstract

One major challenge for semantic segmentation in real-world scenarios is only limited pixel-level labels available due to high expense of human labor though a vast volume of video data is provided. Existing semi-supervised methods attempt to exploit unlabeled data in model training, but they just regard video as a set of independent images. To better explore semi-supervised segmentation problem with video data, we formulate a semi-supervised video semantic segmentation task in this paper. For this task, we observe that the overfitting is surprisingly severe between labeled and unlabeled frames within a training video although they are very similar in style and contents. This is called inner-video overfitting, and it would actually lead to inferior performance. To tackle this issue, we propose a novel inter-frame feature reconstruction (IFR) technique to leverage the ground-truth labels to supervise the model training on unlabeled frames. IFR is essentially to utilize the internal relevance of different frames within a video. During training, IFR would enforce the feature distributions between labeled and unlabeled frames to be narrowed. Consequently, the inner-video overfitting issue can be effectively alleviated. We conduct extensive experiments on Cityscapes and CamVid, and the results demonstrate the superiority of our proposed method to previous state-of-the-art methods. The code is available at https://github.com/jfzhuang/IFR.

1. Introduction

As a fundamental task in computer vision, image semantic segmentation has benefited numerous downstream applications. However, the training data for semantic segmentation requires pixel-level labeling, which is very expensive and time-consuming. Recently, semi-supervised image segmentation (SSIS) is proposed to train models with a limited number of labeled images and additional unlabeled images. Besides regular supervised learning on labeled images, SSIS methods usually construct extra supervision signals for unlabeled images, *e.g.*, consistency constraint [17, 23] and pseudo label [4, 32]. It is shown that they can bring considerable performance improvement on the challenging datasets, *e.g.*, PASCAL VOC [8].

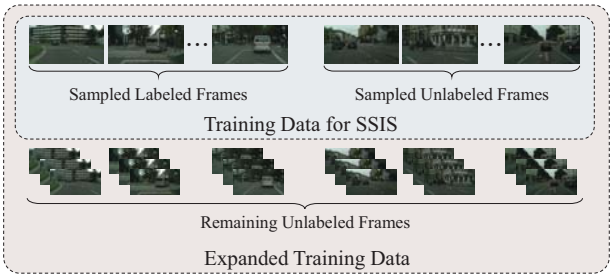

Figure 1. **Illustration of expanded training data**. As commonly adopted in SSIS, only one frame from each video is sampled and used as the training data. These frames are divided into labeled and unlabeled ones. To fully utilize video data, we propose to supplement the remaining video frames into unlabeled data for training.

In real-world scenarios, we can usually collect the video data conveniently and economically, but only a few videos would be annotated with a single frames due to the high expense of human labor. For example, Cityscapes [6] as a representative dataset contains 2975 videos in the training subset, and only the 20th frame in each video is annotated. That is, there are actually many unlabeled frames that can be utilized for model training. However, the existing SSIS methods [4, 17, 23] do not fully exploit video data, in which only one frame is sampled from each video for model training and a large amount of unlabeled frames are ignored. To better explore unlabeled videos, we propose to expand the training data setting of SSIS by leveraging remaining video frames, as shown in Figure 1.

A nature way to utilize extra unlabeled frames is to perform SSIS with adding the remaining frames into unlabeled training data. Here we particularly implement two SOTA SSIS methods, *i.e.*, CAC [17] and CPS [4]. Table 1 gives the

*Corresponding author

Method	Sampled Frames	All Frames
Baseline	66.00	-
+ CAC	69.70	69.80 (+0.10)
Baseline	70.32	-
+ CPS	74.39	74.66 (+0.27)

Table 1. **Performance of SOTA methods with all video frames**. We implement CAC [17] and CPS [4] by adding remaining video frames into unlabeled training data. However, no obvious improvement is gained. Here, Cityscapes with 1/8 labeled data is used, and Deeplabv3+ with the ResNet50 backbone is adopted as segmentation model. The baseline performances are different due to adopting different training settings.

	ID	Accel	+ CAC	+ CPS
T	20	81.5	80.1	79.7
	15	$69.8_{\downarrow 11.7}$	$69.9_{\downarrow 10.2}$	$68.1_{\downarrow 11.6}$
	10	$65.3_{\downarrow 16.2}$	$65.5_{\downarrow 14.6}$	$65.1_{\downarrow 14.6}$
	5	$60.6_{\downarrow 20.9}$	$61.9_{\downarrow 18.2}$	$61.8_{\downarrow 17.9}$
V	20	$49.8_{\downarrow 31.7}$	$51.1_{\downarrow 29.0}$	$51.3_{\downarrow 28.4}$

Table 2. **Performance on Cityscapes-VPS**. T and V represent the training and validation subsets, respectively. ID indicates the frame id. The model is trained on the 20th frame in the training subset. $\downarrow$ represents the accuracy gap comparing to that on the training frames. Performance gap exists not only between two subsets but also within each training video.

results on Cityscapes. It can be seen that there is no obvious performance improvement. The primary reason may be the homogenization of video data, *i.e.*, the contents of different frames within a video are often similar. Given one frame from each video, therefore, the existing semi-supervised methods cannot capture much more information from the remaining frames using a simple extension on training data. Evidently, how to effectively boost the segmentation performance with unlabeled video data is challenging.

Next we first formulate the semi-supervised video segmentation (SSVS) task addressed in this work. There are two main differences comparing to the SSIS task. The first is the *training data*. During training, only one frame sampled from each video is used in SSIS, while all video frames are accessible in SSVS, as shown in Figure 1. The second is the *baseline model*. SSIS focuses on improving image segmentation models, *e.g.*, PSPNet [28], while SSVS concentrates on video segmentation models, *e.g.*, Accel [15]. Since the video models are usually designed for exploiting video characteristics, *e.g.*, feature propagation by utilizing the temporal consistency, we particularly consider the learning method by further exploring video data in this paper.

The overfitting is a key challenge in SSIS, as demonstrated in existing works [17, 23]. Here we investigate the

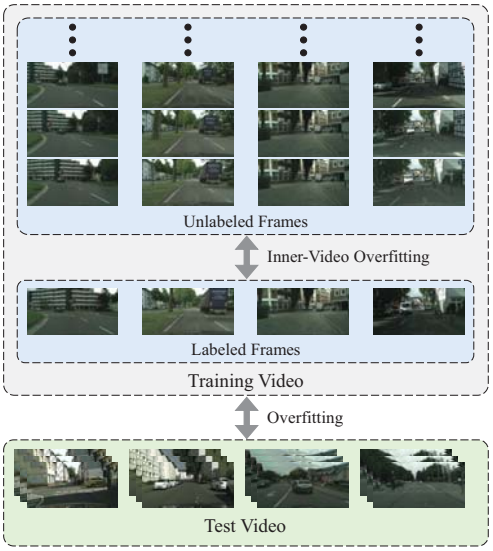

Figure 2. **Illustration of inner-video overfitting**. The commonly concerned overfitting issue is the performance gap between training and test videos. However, we find that overfitting also exists between labeled and unlabeled frames within each training video, which is called inner-video overfitting.

overfitting issue in SSVS with a popular VSS model, *i.e.*, Accel [15]. In particular, we conduct a study experiment on the Cityscapes-VPS dataset [16]. The dataset contains 500 videos, where each video contains 30 frames and every 5 frame is annotated. We randomly select 100 videos for training and the remaining are for validation. To be specific, only the annotation on the 20th frame is used for each video during training. We evaluate the model on both the training and validation subsets, as shown in Table 2. From the results, we have two observations. First, the performance gap exists between the training and validation frames, which is the commonly concerned overfitting issue. Second, the gap also occurs between the training and other frames within the training videos, though they have no significant visual difference as shown in Figure 2. We called this phenomenon as *inner-video overfitting*.

In this work, we argue that the inner-video overfitting is mainly caused by the lack of accurate supervision signals for unlabeled frames, which results in inconsistent performance. Specifically, the model trained on the labeled frames is supervised by the ground-truth labels, which can provide accurate semantic signals. But the model trained on unlabeled frames is either not considered as in Accel or supervised by some constructed signals. Actually, these signals can only provide an indirect constraint like consistency in CAC or noisy semantic supervision like pseudo labels in CPS, which cannot effectively supervise model training on unlabeled frames.

Then a natural question arises: can we use ground-truth

labels to train model on the unlabeled frames? Here we particularly utilize the internal relevance of different frames within a video, as they share similar semantic contents and styles. Following this idea, we propose a novel inter-frame feature reconstruction (IFR) method. The main idea is to reconstruct features of the labeled frame $\mathbf{f}_L$ using the information from features of the unlabeled frame $\mathbf{f}_U$. After that, we apply supervised learning on the reconstructed feature $\mathbf{f}_R$ with the ground-truth label, which actually imposes supervision on $\mathbf{f}_U$ implicitly. During training, $\mathbf{f}_R$ would become similar to $\mathbf{f}_L$ as they are supervised by the same objective. As a result, the feature distribution discrepancy between $\mathbf{f}_L$ and $\mathbf{f}_U$ would be narrowed, and further the inner-video overfitting would be alleviated. In this way, we can also improve the generalization capacity of the model on unseen scenes, *i.e*., testing data. In one word, IFR has two main advantages comparing to other methods. First, it uses the ground-truth labels to train model on unlabeled data rather than the generated pseudo labels [4] or consistency constraint [17, 23], which can provide much more accurate semantic supervision. Second, it collaborates the training on labeled and unlabeled data through the same objective, which are processed separately in the existing methods.

We experimentally evaluate the proposed method on the Cityscapes and CamVid datasets. The results validate the effectiveness of our IFR in alleviating the inner-video overfitting issue, and it can bring a significant performance improvement for mainstream video semantic segmentation methods. The contributions of this work are summarized as follows.

- We formulate the semi-supervised video semantic segmentation task and discover the inner-video overfitting issue is one of the main challenges damaging the performance of SSVS.
- We propose a novel inter-frame feature reconstruction method to alleviate the inner-video overfitting and further boost the performance. IFR essentially utilizes the internal relevance of different frames within a video.
- We experimentally evaluate the effectiveness of our proposed method, and the results on Cityscapes and CamVid demonstrate the superiority of our method to previous state-of-the-art methods.

2. Related Work

Image semantic segmentation. Image semantic segmentation targets to assign each pixel in scene images a semantic class, which is a fundamental yet rather challenging task. Modern deep learning methods for semantic segmentation are mainly based on fully convolutional network (FCN) [20]. FCN firstly uses convolutional layers to replace fully-connected layers and can achieve better performance. To further enhance segmentation results, the dilation convolution [3], pyramid pooling [28], and attention mechanism [10, 13, 29] are proposed to model object relationships and aggregate context information. Besides, HRNet [25] is designed to maintain high resolution feature maps. With the development of transformer, some transformer-based segmentation models [5, 19, 26] are proposed and outperform current CNN-based networks. Despite the success of these models, they are often impeded in practical deployment due to requiring sufficient pixel-wise annotations for learning.

Semi-supervised semantic segmentation. To achieve good representation learning with limited annotations, semi-supervised semantic segmentation is studied by exploring unlabeled data. Existing methods can be roughly divided into three families. The adversarial based methods, *e.g*., AdvSemiSeg [14] and S4GAN [21], utilize a discriminator to distinguish the confidence maps from labeled and unlabeled data predictions. The consistency based methods enforce the consistency of the predictions or intermediate features with various perturbations. The perturbations can be conducted on input images, *e.g*., CutMix [9], ClassMix [22], and CAC [17], and feature space, *e.g*., CCT [23]. The self-learning based methods generate pseudo segmentation maps on unlabeled data. [2, 11, 27] propose to generate pseudo labels in an offline manner and retrain the model iteratively. While PseudoSeg [32] and CPS [4] follow the FixMatch [24] scheme and design an online pseudo labeling mechanism. Note that Naive-Student *et al*. [2] proposes to leverage self-learning in extra unlabeled video sequences to improve the performance of full-supervised image segmentation models. Different from that, our work focuses on improving representation learning of video segmentation models with limited annotations.

However, existing methods are not designed for video data, which only regards unlabeled videos as a collection of independent images. As shown in Table 1, more extra frames cannot bring considerable improvement due to severe data homogeneity. In this paper, we propose a novel approach to explore the internal relevance of video data.

Video semantic segmentation. Video semantic segmentation aims to predict pixel-level semantics for each video frame. Different from static images, videos embody rich temporal information that can be exploited to improve semantic segmentation performance. DFF [30] firstly proposes feature propagation to reuse key frame features under the guidance of estimated optical flows, which can reduce the average computational cost. Inspired by DFF, Accel [15] proposes an adaptive fusion policy to effectively integrate the predictions from the key and current frames. DAVSS [31] proposes to correct the distorted features caused by inaccurate optical flow when propagating

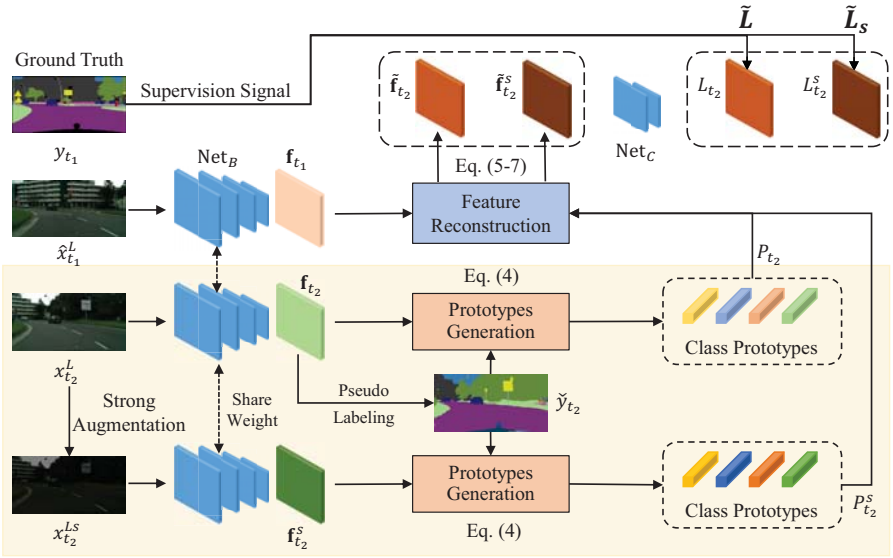

Figure 3. **Overview of Inter-Frame Feature Reconstruction**. For a labeled video, we sample the annotated frame $\hat{x}^L_{t_1}$, a random frame $x^L_{t_2}$ and the given label y_{t_1} to construct a training sample. t_1 and t_2 represent different timestamps. During training, after feature extraction, we generate class prototypes on $\mathbf{f}_{t_2}$ and $\mathbf{f}^s_{t_2}$ based on pseudo label $\check{y}_{t_2}$. Then, prototypes P_{t_2} and $P^s_{t_2}$ are used to reconstruct $\mathbf{f}_{t_1}$, resulting in $\widetilde{\mathbf{f}}_{t_2}$ and $\widetilde{\mathbf{f}}^s_{t_2}$. Finally, cross entropy loss $\widetilde{L}$ and $\widetilde{L}_s$ are calculated on logits L_{t_s} and $\widetilde{L}_{t_s}$ with y_{t_1}, respectively. To be notice, IFR can also be applied on unlabeled videos by only replacing label y_{t_1} with pseudo label $\check{y}_{t_1}$ of the sampled frame. Best viewed in color and zoom in.

features. Besides the feature propagation paradigm, some recent works propose to improve the performance of light-weight models with temporal constraints [7, 18] and attention mechanism [12].

Different from the image-based methods, video segmentation methods can use unlabeled frames nearby the labeled frame for training. However, existing methods still require sufficient annotated frames, and other unlabeled videos are not fully explored. To the best of our knowledge, this work is the first attempt to design a semi-supervised learning method for video segmentation methods.

3. Method

3.1. Problem Definition

In this work, we focus on the semi-supervised video semantic segmentation problem. Formally, assume a small set of labeled videos with one frame annotated and a large set of unlabeled videos are provided. Let $\mathcal{V} = \{x_1, \cdots, x_l\}$ represents l frames in a video with x_i as the ith frame with spatial resolution of $H \times W$. Let $\mathcal{D}_L = \{(\mathcal{V}^L_1, \hat{x}_1, y_1), \cdots, (\mathcal{V}^L_{n_L}, \hat{x}_L, y_{n_L})\}$ represents the n_L labeled videos, where $\hat{x}_i$ is the annotated frame of the ith video, $y_i \in \mathbb{R}^{C\times H\times W}$ corresponds to the pixel-level one-hot label, and C is the number of classes. Let $\mathcal{D}_U = \{\mathcal{V}^U_1, \cdots, \mathcal{V}^U_{n_U}\}$ represents the n_U unlabeled videos. Besides, another set of labeled videos $\mathcal{D}_V = \{(\mathcal{V}^V_1, \hat{x}_1, y_1), \cdots, (\mathcal{V}^V_{n_V}, \hat{x}_{n_V}, y_{n_V})\}$ is used for performance evaluation.

Our work aims to learn a segmentation model from $\mathcal{D}_L$ and $\mathcal{D}_U$, and generalize it to $\mathcal{D}_V$. Generally, a segmentation network $\texttt{Net} = \texttt{Net}_B \circ \texttt{Net}_C$ can be regarded as a composition of the backbone $\texttt{Net}_B$ for feature extraction and the classifier $\texttt{Net}_C$ for semantic prediction. Following previous methods [4, 9, 17, 22, 23, 32], the objective of semi-supervised learning can generally be summarized as two loss functions. The first one is a regular cross-entropy loss on the labeled data:

$$L_{sup} = -\mathbb{E}_{(\hat{x},y)\in\mathcal{D}_L} \sum_{i=1}^{H\times W} \sum_{c=1}^{C} y^{(i,c)} \log p^{(i,c)}, \quad (1)$$

where $p = \texttt{Net}(\hat{x})$ and $p^{(i,c)}$ represents the softmax probability of the pixel i belonging to the cth class. The second loss aims to train model on unlabeled data with some constructed supervision signals, *e.g.*, consistency constraint or pseudo label, which is denoted by L_{unsup} in this paper. Then the overall training objective can be presented as follows

$$L = L_{sup} + \lambda L_{unsup}, \quad (2)$$

where λ is a trade-off parameter.

3.2. Inter-Frame Feature Reconstruction

In this work, we mainly focus on addressing the inner-video overfitting and present a novel inter-frame feature reconstruction (IFR) method. Instead of leveraging the labeled and unlabeled data via different loss functions, IFR

involves unlabeled frames in supervised learning of labeled frames and consequently can narrow the feature distribution discrepancy between different frames within a video. The overview of our solution is illustrated in Figure 3. To make it clear, we first elaborate on the core component of our method on labeled video data, and then explain two extensions to strong augmented data and unlabeled video data.

Specifically, we sample an annotated frame $\hat{x}^L_{t_1}$ with the label y_{t_1} and an unlabeled frame $x^L_{t_2}$ from a labeled video to construct a training sample, denoted by $(x_{t_1}, x_{t_2}, y_{t_1})$, where the superscripts L and $\hat{}$ are omitted to simplify the notations. Firstly, we extract the features on x_{t_1} and x_{t_2} with a shared backbone, resulting in $\mathbf{f}_{t_1}$ and $\mathbf{f}_{t_2}$. The key idea of our IFR is to reconstruct $\mathbf{f}_{t_1}$ with the information of $\mathbf{f}_{t_2}$ and then conduct supervised learning with the given labels on the reconstructed features. Since different frames within a video contain similar semantic content, we consider using the class prototypes generated from $\mathbf{f}_{t_2}$ for feature reconstruction. Here the class prototypes refer to the class-wise feature centroids that commonly represent class semantics. Specifically, we calculate the pseudo label $\check{y}_{t_2}$ of $\mathbf{f}_{t_2}$ as

$$\check{y}_{t_2} = \texttt{argmax}(\texttt{Net}_C(\mathbf{f}_{t_2})). \tag{3}$$

Then, we group the pixel-wise features in $\mathbf{f}_{t_2}$ belonging to the same class according to $\check{y}_{t_2}$, *i.e.*,

$$P^{(c)}_{t_2} = \frac{\sum_i \mathbf{f}^{(p)}_{t_2} * \mathbb{1}(\check{y}^{(i,c)}_{t_2} == 1)}{\sum_i \mathbb{1}(\check{y}^{(i,c)}_{t_2} == 1)}, \tag{4}$$

where $\mathbb{1}$ is an indicator function and $\check{y}^{(i,c)}_{t_2}$ represents the one-hot label of pixel i belonging to the cth class. To represent $\mathbf{f}_{t_1}$ with the generated class prototypes P_{t_2}, we consider adopting the commonly used attention mechanism

$$s^{(i,c)} = \bar{\mathbf{f}}^{(i)T}_{t_1} \bar{P}^{(c)}_{t_2}, \tag{5}$$

$$s^{(i,c)} = \frac{e^{s^{(i,c)}/\tau_{re}}}{\sum_{c=1}^{C} e^{s^{(i,c)}/\tau_{re}}}, \tag{6}$$

$$\widetilde{\mathbf{f}}^{(i)}_{t_2} = \sum_{c=1}^{C} s^{(i,c)} P^{(c)}_{t_2}, \tag{7}$$

where $\bar{\mathbf{f}^{(i)}} = \mathbf{f}^{(i)}/\|\mathbf{f}^{(i)}\|$ is the L2 normalization. Here we adjust the softmax operation by dividing by a temperature τ_{re}.

After that, we generate the classification probabilities $\widetilde{p} = \sigma(\texttt{Net}_C(\widetilde{\mathbf{f}}_{t_2}))$, where σ indicates a softmax function. To perform supervised learning similar to labeled frames, we adopt the cross-entropy with the given label y_{t_1} as follows

$$\widetilde{L} = -\mathbb{E} \sum_{i=1}^{H\times W} \sum_{c=1}^{C} y^{(i,c)}_{t_1} \log \widetilde{p}^{(i,c)}. \tag{8}$$

Then the overall training objective can be presented as

$$L = L_{sup} + \lambda \widetilde{L}. \tag{9}$$

During training, $\widetilde{\mathbf{f}}_{t_2}$ would become similar to $\mathbf{f}_{t_1}$ since they are enforced by the same loss function and supervision signal. Therefore, the model is encouraged to narrow the distribution between $\mathbf{f}_{t_2}$ and $\mathbf{f}_{t_1}$, *i.e.*, keep the feature consistency of different frames within a video.

3.3. Extended Solution

Strong Augmented Data. Existing works [9, 22, 23, 27] have shown that strong data augmentation on unlabeled data can effectively improve model generalization in semi-supervised learning. As shown in the lower half of Figure 3, we extend the IFR training on strong augmented frames. Specifically, taking the labeled video data as an example, we impose strong augmentation, *e.g.*, color jitter, on $x^L_{t_2}$ to get $x^{Ls}_{t_2}$. After feature extraction, we also use $\mathbf{f}^s_{t_2}$ to perform feature reconstruction. Differently, when generating the prototypes, we use the pseudo label of $\mathbf{f}_{t_2}$ rather than $\mathbf{f}^s_{t_2}$ for more accurate semantic prediction. After that, a regular IFR training procedure is adopted with $\widetilde{L}^s_L$. Essentially, we impose an implicit consistency constraint on $\mathbf{f}_{t_2}$ and $\mathbf{f}^s_{t_2}$, since we narrow their distribution discrepancy with $\mathbf{f}_{t_1}$. Finally, the overall training objective can be presented as

$$L = L_{sup} + \lambda(\widetilde{L} + \lambda_s \widetilde{L}^s), \tag{10}$$

where λ_s is an extra control parameter.

Unlabeled Video Data. By far, our proposed IFR is only applied to labeled videos. To further explore a large amount of unlabeled videos, we make a simple extension to the IFR solution. Specifically, we randomly sample two frames $x^U_{t_1}$ and $x^U_{t_2}$ from each unlabeled video. Similarly, strong augmentation is applied on $x^U_{t_2}$, resulting in $x^{Us}_{t_2}$. Since no frame label is provided, we need to build supervision signal for reconstructed features. In particular, we adopt the pseudo label $\check{y}_{t_1}$ of $x^U_{t_1}$. Then a training sample $(x^U_{t_1}, x^U_{t_2}, x^{Us}_{t_2}, \check{y}_{t_1})$ is obtained and naturally it can be used in the IFR training procedure as for the labeled samples. Finally, the overall training objective can be presented as

$$L = L_{sup} + \lambda_L(\widetilde{L}_L + \lambda_s \widetilde{L}^s_L) + \lambda_U(\widetilde{L}_U + \lambda_s \widetilde{L}^s_U), \tag{11}$$

where subscripts L and U represent loss terms calculated on the labeled and unlabeled videos, respectively. λ_L and λ_U correspond to their trade-off parameters.

4. Experiments

4.1. Dataset

Cityscapes [6] is a representative dataset in semantic segmentation and autonomous driving domain. It focuses

Method	1/30 (100)	1/16 (186)	1/8 (372)	1/4 (744)	1/2 (1488)
Accel	45.73	52.10	57.12	60.55	62.83
+ CCT [23]	48.05 (+2.32)	53.25 (+1.15)	58.88 (+1.76)	62.00 (+1.45)	64.02 (+1.19)
+ CAC [17]	48.83 (+3.10)	54.56 (+2.46)	58.55 (+1.43)	62.78 (+2.23)	63.87 (+1.04)
+ CPS [4]	48.97 (+3.24)	54.69 (+2.59)	58.97 (+1.85)	62.43 (+1.88)	63.74 (+0.91)
+ Ours	**52.86 (+7.13)**	**56.39 (+4.29)**	**60.08 (+2.96)**	**63.45 (+2.90)**	**64.53 (+1.70)**

Table 3. **Comparison with state-of-the-art methods on the Cityscapes validation subset** under different partition protocols. Accel is adopted as the supervised baseline that is only trained on labeled data. Our method gets more gains especially for few labeled training data.

Method	1/30 (15)	1/16 (29)	1/8 (58)	1/4 (117)	1/2 (234)
Accel	42.37	47.57	50.78	56.40	59.37
+ CCT [23]	47.09 (+4.72)	52.45 (+4.88)	54.50 (+3.72)	58.69 (+2.29)	61.58 (+2.21)
+ CAC [17]	46.85 (+4.48)	52.16 (+4.59)	54.03 (+3.25)	59.67 (+3.27)	63.17 (+3.80)
+ CPS [4]	46.05 (+3.68)	52.04 (+4.47)	55.30 (+4.52)	59.02 (+2.62)	62.49 (+3.12)
+ Ours	**49.50 (+7.13)**	**53.71 (+6.14)**	**57.37 (+6.59)**	**61.27 (+4.87)**	**63.86 (+4.49)**

Table 4. **Comparison with state-of-the-art methods on the CamVid test subset** under different partition protocols. Accel is adopted as the supervised baseline that is only trained on labeled data. Our method gets more gains especially for few labeled training data.

on semantic understanding to urban street scenes. The training and validation subsets contain $2,975$ and 500 videos, respectively, and each video contains 30 frames at a resolution of 1024×2048. The 20th frame in each is annotated by pixel-level semantic labels with 19 categories.

CamVid [1] also focuses on the semantic understanding to urban street scenes, but it contains less data than Cityscapes. It has four driving videos and each video contains frames ranging from $3,600$ to $11,000$ at a resolution of 720×960. Every 30th frame of videos is annotated with 11 semantic classes, which results in a total of 701 samples. Similar to Cityscapes, we split videos into 701 videos and each video contains 30 frames. All videos are divided into the trainval set with 468 videos and test set with 233 videos.

We follow the partition protocols of CutMix [9] and CPS [4] and divide the whole training set via randomly sub-sampling 1/2, 1/4, 1/8, 1/16, and 1/30 of all training videos, *i.e.*, 2975 videos in Cityscapes and 468 videos in CamVid, as the labeled set and regard the remaining videos as the unlabeled set. In the implementation, we follow [2] to add the labeled set into the unlabeled set for unsupervised learning, which can slightly improve performance. Following [9, 14, 21–23], we apply bilinear interpolation to resize every video frame in Cityscapes and CamVid to $512 * 1024$ and $360 * 480$ for the efficiency of training and inference, respectively.

4.2. Implementation Details

Here we particularly adopt Accel [15] as video semantic segmentation architecture due to its good performance. It consists of two segmentation branches, *i.e.*, a heavy reference branch and a light-weight update branch, an optical flow network, and a score fusion layer. To be consistent with previous works [9,17,23], we adopt PSPNet [28] to implement the segmentation branches. Specifically, ResNet-101 and ResNet-18 are adopted as the backbone of reference and update branches, respectively.

Accel uses a two-stage training procedure. In stage one, two segmentation branches are trained separately on a specific dataset, *e.g.*, Cityscapes [6]. To improve representation learning with limited annotations, we apply semi-supervised methods to this stage. The segmentation model is trained using a SGD optimizer with a momentum of 0.9 and a weight decay of 10^{-4}. The learning rate is set at 10^{-3} for the backbone parameters and 10^{-2} for others, which is annealed following the poly learning rate policy. In stage two, two segmentation branches are fixed, and the classifier is jointly trained with the optical flow network and the score fusion layer by following the standard supervised learning paradigm. The training settings follow the original Accel implementation. Actually, the training in the stage two also suffers from limited annotations. However, how to design a suitable semi-supervised learning method for optical flow network is beyond the scope of this work, which can be explored in the future.

We evaluate the segmentation performance on the validation videos, *i.e.*, validation subset in Cityscapes and test set in CamVid. Following Accel, for each test video, we conduct the reference branch on a selected key frame and update branch on the annotated frame. The segmentation results are predicted via the feature propagation and score fusion. We evaluate different methods with mean Intersection-over-Union (mIoU) as metric. The key frame interval is set as 5 throughout the experiments.

For our IFR, there are three trade-off hyperparameters in Eq. 11 and one temperature in Eq. 6, *i.e.*, λ_L, λ_U, λ_s and τ_{re}. We set $\lambda_L = 0.01$, $\lambda_U = 0.001$, $\lambda_s = 0.1$ and $\tau_{re} = 0.5$ for all experiments.

4.3. Performance Comparison

To demonstrate the superiority of our method, we make a comparison with recent state-of-the-art methods, *i.e.*, two consistency based SSIS methods including CCT [23] and CAC [17], and one self-learning based SSIS method CPS [4]. However, it is hard to directly compare with these methods since they are implemented under different settings, *e.g.*, segmentation models, data splits, and training settings. Besides, these methods are not applied to our used video semantic segmentation model Accel. Therefore, we reproduce these methods according to their official codes, where all of them are equipped with the same base segmentation model, *i.e.*, PSPNet, and trained with the same data splits and training settings (*i.e.*, optimizer and hyperparameters). In addition, different data augmentations are adopted in existing works, *e.g.*, random cropping and flipping in [23], and extra random scaling in [17] and cutmix in [4]. For fair comparison, we adopt the same random cropping and flipping for supervised learning of all comparison methods while keeping their original implementations for unsupervised learning. In this way, we can fairly compare the improvement brought by different methods on the basis of the same baseline with supervised learning.

The comparison on Cityscapes and CamVid is shown in Table 3 and Table 4. From the results, we have the following two observations. First, our IFR can bring significant performance improvement under all partition protocols comparing to the baseline that only uses the labeled data. A larger gain is achieved for the cases with less labeled data, *e.g.*, 7.13% mIoU gain with 100 samples in Cityscapes and 7.14% mIoU gain with 15 samples in CamVid. Such experimental results well verify that IFR can effectively improve the generalization of models. Second, our method surpasses other state-of-the-art methods, including CCT [23], CAC [17], and CPS [4], by a large margin. For example, it outperforms CPS by 3.98% and 3.46% under 1/30 partition protocol on Cityscapes and CamVid, respectively. It shows that IFR can better utilize the unlabeled video data in training model.

4.4. Ablation Study

In this subsection, we conduct experiments to reveal the effectiveness of our proposed method. All experiments are particularly conducted with 1/30 labeled data on Cityscapes. For efficient training, we adopt PSPNet with ResNet18 as the segmentation network by default.

Effect of Components. To reveal the contribution of our proposed components, we conduct an extensive study by evaluating their combinations, and the results are shown in Table 5. It can be seen that each component can bring performance improvement comparing to the baseline. In particular, the feature reconstruction on both the labeled and unlabeled frames can achieve higher accuracy than the setting only equipped with each of them. Finally, the combination of all components performs best.

L_{sup}	$\widetilde{L}_L$	$\widetilde{L}_U$	$\widetilde{L}_L^s + \widetilde{L}_U^s$	mIoU (%)
√				43.68
√	√			46.13 (+2.45)
√		√		46.81 (+3.13)
√			√	46.66 (+2.98)
√	√	√		47.51 (+3.83)
√	√	√	√	**48.40 (+4.72)**

Table 5. **Ablation study on our proposed components**. Each component can bring performance improvement comparing to the baseline, and their combination performs best.

Method	Stage One		Stage Two
	PSPNet18	PSPNet101	
Baseline	43.68	50.24	45.73
+ CCT	45.60 (+1.92)	52.60 (+2.36)	48.05 (+2.32)
+ CAC	46.50 (+2.82)	53.50 (+3.26)	48.83 (+3.10)
+ CPS	46.81 (+3.13)	53.26 (+3.02)	48.97 (+3.24)
+ Ours	**48.40 (+4.72)**	**54.40 (+4.16)**	**52.86 (+7.13)**

Table 6. **Ablation study on improvement in multi-stage training**. Following Accel, stage one involves the training of image segmentation model while stage two mainly involves the training of optical flow network and score fusion. Comparing to others, our method can bring more gains in both training stages.

Improvement in Two Stages. The used Accel is trained in a two-stage manner. Here we investigate the performance improvement in the two-stage training, and the results are shown in Table 6. In stage one, only image segmentation models, *i.e.*, PSPNet18 and PSPNet101, are involved. Comparing to other methods, our method can obtain more gain over the baseline due to better utilization of unlabeled videos. In stage two, the training of optical flow network and score fusion layer are involved, *i.e.*, no semi-supervised method is introduced in this stage. However, we observe an interesting phenomenon that our trained segmentation model can further bring performance improvement over the baseline, while other models only almost maintain improvement brought in stage one. It is because our method can help the segmentation model to extract features with a similar distribution for different frames, which is important for the feature fusion in Accel.

Performance with Different VSS Architectures. To study the generalization ability of our method, we further apply it

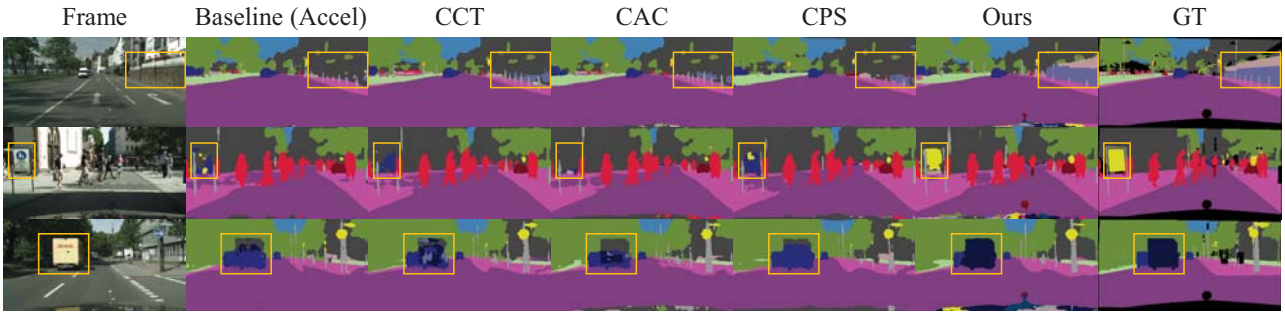

Figure 4. **Qualitative results comparison** on the Cityscapes dataset using 1/30 labeled samples. The proposed IFR produces better results than the baseline and other methods. We highlight the details with the yellow boxes. Best viewed in color and zoom in.

Method	Baseline	+ Ours
DFF [30]	44.19	49.97 (+5.78)
DAVSS [31]	46.51	51.97 (+5.46)

Table 7. **Performance on different VSS architectures**. Our method can bring significant improvement for different video semantic segmentation architectures consistently.

	ID	Accel	+ CAC	+ CPS	+ Ours
T	20	81.5	80.1	79.7	76.5
	15	$69.8_{\downarrow 11.7}$	$69.9_{\downarrow 10.2}$	$68.1_{\downarrow 11.6}$	$71.0_{\downarrow 5.5}$
	10	$65.3_{\downarrow 16.2}$	$65.5_{\downarrow 14.6}$	$65.1_{\downarrow 14.6}$	$67.4_{\downarrow 9.1}$
	5	$60.6_{\downarrow 20.9}$	$61.9_{\downarrow 18.2}$	$61.8_{\downarrow 17.9}$	$66.8_{\downarrow 9.7}$
V	20	$49.8_{\downarrow 31.7}$	$51.1_{\downarrow 29.0}$	$51.3_{\downarrow 28.4}$	$53.9_{\downarrow 22.6}$

Table 8. **Performance on Cityscapes-VPS**. T and V represent training and validation subsets, respectively. ID represents for frame id. Model is trained on the 20th frame in training subset. $\downarrow$ represents the accuracy gap comparing to trained frames. Obviously, our method can significantly reduce both overfitting and inner-video overfitting issues.

to different video semantic segmentation architectures. We particularly adopt two widely adopted video segmentation architectures, *i.e.*, DFF [30] and DAVSS [31]. As shown in Table 7, our proposed method can bring significant performance improvement consistently.

Effectiveness on Alleviating Overfitting. To verify the effectiveness of our method on tackling the overfitting, we conduct an analysis experiment on Cityscapes-VPS, in which each video is annotated with multiple frames, and the results are shown in Table 8. Comparing to the baseline, it can be seen that our method can significantly reduce the accuracy gap not only between training and validation videos but also within training videos. It verifies that our method can effectively alleviate the overfitting issue in video semantic segmentation.

Effectiveness on Temporal Consistency. It is important for VSS methods to produce temporally stable predictions. Thus, we further verify the effectiveness of our method on improving temporal consistency. Particularly, we follow [18] and adopt temporal consistency (TC) score as the evaluation metric. As shown in Table. 9, our method can bring a significant improvement, since consistent predictions among different frames can be achieved.

Method	Accel	+ CCT	+ CAC	+ CPS	+ Ours
TC	70.04	71.43	71.47	70.74	73.88

Table 9. **Comparison on Temporal Consistency**. We evaluate different methods with temporal consistency (TC) score [18]. Our proposed method can bring a significantly improvement.

Qualitative Results. Figure 4 presents the visual comparison with the baseline and other semi-supervised methods. We can observe that the results of our method are commonly superior to others.

5. Conclusion

In this paper, we focus on the semi-supervised video semantic segmentation problem, and propose a novel inter-frame feature reconstruction approach. Our IFR exploits a large number of unlabeled frames by cooperating them into supervised learning of labeled frames, which essentially narrows the feature distribution of different frames within a video. Furthermore, we propose two extensions on strong augmented data and unlabeled videos. IFR can effectively alleviate the inner-video overfitting, and extensive experiments on Cityscapes and CamVid validate the effectiveness of our method, which outperforms previous state-of-the-art methods.

Acknowledge

This work is supported by the National Natural Science Foundation of China under Grant No.62176246 and No.61836008. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009. 6

[2] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *ECCV*, 2020. 3, 6

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 3

[4] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*, 2021. 1, 2, 3, 4, 6, 7

[5] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv preprint arXiv:2107.06278*, 2021. 3

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 5, 6

[7] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: joint learning of video segmentation and optical flow. In *AAAI*, 2020. 4

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1

[9] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *BMVC*, 2019. 3, 4, 5, 6

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 3

[11] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *ICCV*, 2021. 3

[12] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, 2020. 4

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

[14] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *BMVC*, 2018. 3, 6

[15] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *CVPR*, 2019. 2, 3, 6

[16] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, 2020. 2

[17] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In *CVPR*, 2021. 1, 2, 3, 4, 6, 7

[18] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *ECCV*, 2020. 4, 8

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, 2021. 3

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

[21] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *TPAMI*, 2019. 3, 6

[22] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *WACV*, 2021. 3, 4, 5, 6

[23] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 7

[24] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 3

[25] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020. 3

[26] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 3

[27] Jianlong Yuan, Yifan Liu, Chunhua Shen, Zhibin Wang, and Hao Li. A simple baseline for semi-supervised semantic segmentation with strong data augmentation. In *ICCV*, 2021. 3, 5

[28] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2, 3, 6

[29] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018. 3

[30] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 3, 8

[31] Jiafan Zhuang, Zilei Wang, and Bingke Wang. Video semantic segmentation with distortion-aware feature correction. *TCSVT*, 2020. 3, 8

[32] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*, 2020. 1, 3, 4