WeClick: Weakly-Supervised Video Semantic Segmentation with Click Annotations

Peidong Liu1,*, Zibin He1,*, Xiyu Yan1,*, Yong Jiang1,2, Shu-Tao Xia1,2,†
Feng Zheng3, Maowei Hu1,4
1Tsinghua Shenzhen International Graduate School, Tsinghua University
2PCL Research Center of Networks and Communications, Peng Cheng Laboratory
3Department of Computer Science and Engineering, Southern University of Science and Technology
4Shenzhen Rejoice Sport Tech. Co., LTD

{lpd19, hzb19, yanqy17}@mails.tsinghua.edu.cn, {jiangy, xiast}@sz.tsinghua.edu.cn, zhengf@sustech.edu.cn, humaowei@51yund.com

ABSTRACT
Compared with tedious per-pixel mask annotating, it is much easier to annotate data by clicks, which costs only several seconds for an image. However, applying clicks to learn video semantic segmentation model has not been explored before. In this work, we propose an effective weakly-supervised video semantic segmentation pipeline with click annotations, called WeClick, for saving laborious annotating effort by segmenting an instance of the semantic class with only a single click. Since detailed semantic information is not captured by clicks, directly training with click labels leads to poor segmentation predictions. To mitigate this problem, we design a novel memory flow knowledge distillation strategy to exploit temporal information (named memory flow) in abundant unlabeled video frames, by distilling the neighboring predictions to the target frame via estimated motion. Moreover, we adopt vanilla knowledge distillation for model compression. In this case, WeClick learns compact video semantic segmentation models with the low-cost click annotations during the training phase yet achieves real-time and accurate models during the inference period. Experimental results on Cityscapes and Camvid show that WeClick outperforms the state-of-the-art methods, increases performance by 10.24% mIoU than baseline, and achieves real-time execution.

CCS CONCEPTS
• Computing methodologies → Video segmentation.

KEYWORDS
video semantic segmentation, weakly-supervised learning, click annotations, knowledge distillation

† Corresponding author.
* Equally contributed.
This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204 and the National Natural Science Foundation of China under Grant 61771273, 61972188.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ACM Reference Format:
Peidong Liu1,*, Zibin He1,*, Xiyu Yan1,*, Yong Jiang1,2, Shu-Tao Xia1,2,†, Feng Zheng3, Maowei Hu1,4 . 2021. WeClick: Weakly-Supervised Video Semantic Segmentation with Click Annotations. In Proceedings of the 29th ACM International Conference on Multimedia (MM ’21), October 20–24, 2021, Virtual Event, China. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475217

1 INTRODUCTION
Video semantic segmentation (VSS), one of the fundamental high-level tasks in computer vision, aims to assign a semantic label to each pixel in every video frame. Compared with image semantic segmentation, VSS has a wider range of application scenarios, e.g. autonomous driving and human-computer interaction [59], which deserves more attention from researchers.

In recent years, the community has witnessed substantial progress in VSS. However, VSS still encounters major bottlenecks because most existing VSS methods [12, 30, 39, 66] depend on large-scale per-pixel masks to learn sufficient semantic information for obtaining well-performed models. Annotating per-pixel masks is time-consuming because it takes an annotator 1.5 hours on average to label a frame. Therefore, Cityscape dataset [8] annotates only 1 frame for a 30-frame video snippet to reduce huge human effort.

To cope with the fine-annotation scarcity issue, a natural way is to use weakly-supervised annotations to train VSS networks. Researchers have explored several weakly-supervised methods in image semantic segmentation, including click-based [1, 40], box-based [9, 23, 25], and extreme point-based [38, 55] approaches. Among these weakly-supervised approaches, click-based methods reduce the tedious labeling process to the greatest extent because it costs the annotator only several seconds for an image. Figure 1(a) and (b) shows the difference between per-pixel masks and click annotations. However, to the best of our knowledge, little attention has been paid to the weakly-supervised learning algorithms with click annotations in VSS. In order to promote the development of this field, we extend the classical click-based learning method [40] in image semantic segmentation to video semantic segmentation for learning semantic information with low-cost click labels in video data, which is named the weakly training scheme in this work.

However, directly training with click annotations leads to poor segmentation performance because detailed semantic information is not captured by clicks. Inspired by the previous self-distillation work [21, 64], to properly exploit temporal information in plenty of unlabeled video frames, we design a novel temporal self-distillation
Figure 1: Examples of (a) fine annotations and (b) click annotations for an image from Cityscapes validation set. (c) and (d) are the semantic segmentation results trained with fine annotations and click annotations, respectively. It can be observed that (d) achieves comparable segmentation results with (c) in some instance class, e.g., car, person, rider.

Figure 2: Comparison on the accuracy and inference speed of our method and the state-of-the-art weakly-supervised methods: Gated CRF [40] (click-based image semantic segmentation) and ADVENT [52] (unsupervised domain adaptation image semantic segmentation). The experiment is conducted on Cityscapes with an Nvidia Tesla V100 GPU.

2 RELATED WORK

2.1 Video Semantic Segmentation

Video semantic segmentation (VSS), also known as video scene parsing (VSP), refers to the process of allotting a semantic label for each pixel in every video frame [59]. VSS is greatly different from image semantic segmentation in that the former is characterized by abundant temporal information. According to the usage of optical flow [32, 66], previous VSS works are summarized into two categories. The first one is to accelerate VSS algorithms by propagating the predictions of previous frames to the target frame based on the frame-pair input. Then the neighboring predictions are warped to the target frame with the estimated optical flow to supervise the learning of the target frame. Besides, a well-designed consistency matrix is employed to regularize the propagation of the estimated motion by adaptively tuning the importance of memory flow in the MFD training scheme.

In this paper, we propose an effective weakly-supervised video semantic segmentation pipeline with click annotations, called WeClick, which learns models with low-cost click labels in the training phase yet achieves fast and accurate prediction without any additional computation cost and post-processing during inference. In specific, WeClick applies the weakly training scheme to alleviate the issue of fine-annotation scarcity and the MFD training scheme to exploit the temporal knowledge. Moreover, to further ease the inference latency problem, WeClick adopts vanilla knowledge distillation for model compression. After training the compact model, all the teacher net and the motion estimation net in the training phase are removed. Benefiting from the proposed training schemes, as Figure 1(c) and (d) shows, WeClick achieves comparable segmentation results with the models trained with fine annotations. During inference, WeClick only keeps the student net as the VSS model with single-frame input so that the proposed training schemes boost the model accuracy without additional computation cost. As Figure 2 illustrates, our DeeplabV3+ with MobileNetV2 can reach higher accuracy (i.e., mIoU) with a faster inference speed compared with other state-of-the-art weakly-supervised methods.

Extensive experiments on benchmark Cityscapes and Camvid demonstrate the effectiveness and generalization of our method. WeClick outperforms the state-of-the-art methods based on image-level and click-level annotations and the most significant improvement in terms of mIoU reaches 10.24% than baseline on Cityscapes. Several lightweight backbones including ResNet18, ResNet34 [16] and MobileNetV2 [48], empirically verify that WeClick improves segmentation accuracy with a faster inference speed.

In summary, our main contributions are as follows:

- We propose the WeClick pipeline, the first study to explore click-based learning in VSS, which learns compact models with low-cost click labels in the training phase yet achieves real-time and accurate prediction in the inference period.
- We employ the weakly training scheme, which is extended from the click-based method in image semantic segmentation, to mitigate the fine-annotation scarcity issue in VSS.
- To boost the segmentation performance, we design a novel Memory Flow knowledge Distillation (MFD) scheme, which utilizes memory flow in plentiful unlabeled video frames.
- Empirical experiments on Cityscapes and Camvid show that the compact models with the proposed training schemes outperform other state-of-the-art weakly-supervised VSS methods substantially.
using optical flow. For example, the methods [24, 30, 58, 66] obtains the segmentation result of the next frame by processing the feature map or segmentation mask of the previous frame via optical flow, thus greatly reducing redundant calculations in the video segmentation. However, the accuracy of the segmentation is reduced. The second one is to use optical flow and other modules to fuse the features of the preceding and subsequent frames or add constraints to learn stronger representation ability for higher accuracy of single frame semantic segmentation [12]. Our work belongs to the second category with weakly-supervised annotations.

### 2.2 Weakly-Supervised Segmentation

In recent years, various types of weakly-supervised semantic segmentation techniques have been studied and developed to reduce the demand for large-scale detailed annotations. The typical weakly-supervised forms of image semantic segmentation are image category tagging [46, 53, 54, 56], bounding box annotations [9, 23, 25], scribble annotations [7, 31], point annotations [1, 40] and eye movement annotations [41]. We focus on the weakly-supervised learning methods based on the low-cost click annotations.

In this context, a series of weakly-supervised learning methods based on point labeling are investigated. Firstly, inspired by a human-oriented object, Bearman et al. [1] proposes a supervision mechanism based on point label in each class for training networks. Besides, they prove that the model of point-level supervision training is better than that of the image-level supervision with a fixed annotating budget by manually annotating and evaluating the annotation time on the Pascal VOC 2012 dataset [11, 15]. The above work is the beginning of point labeling supervision. Tang et al. [49] studies a new method on this basis, in order to minimize the performance gap between weakly- and fully-supervised semantic segmentation. They first propose to jointly minimize a partial cross-entropy loss and a regularized loss for labeled pixels and unlabeled pixels respectively. Next, they further extend the regularization loss [50] to a more general loss function, such as graph cuts or dense CRFs. The aforementioned methods somewhat rely on alternative sources of supervision such as pre-training on other datasets. Recently, Obukhov et al. [40] propose Gated CRF loss for the unlabeled pixels. It can be trained easily with the standard Stochastic Gradient Descent algorithm without any pre- and post-processing operations and achieves state-of-the-art performance for click-based image semantic segmentation algorithms. However, the VSS models, which are directly trained with Gated CRF loss, perform poorly for limited semantic information in click annotations. Therefore, we extend Gated CRF loss to the VSS model for click-based learning and further utilize temporal information in plenty of unlabeled video data via the proposed training schemes.

The above methods focus on image semantic segmentation, which is based on the relatively simple images in the Pascal VOC dataset. In fact, VSS in complex scenes has a more expensive annotation cost and needs to be broken through in weakly-supervised methods [3, 5, 12]. For examples, Saleh et al. proposes a weakly-supervised two-stream (WSTS) method for VSP [47], which handles foreground and background objects evenly, with one stream taking the image and the other taking the optical flow to extract features. [27] proposes a weakly-supervised method that uses temporal information to train a network on a video dataset labeled at the image level, which is automatically harvested from the web. The model obtains activated regions from each video frame and then aggregates them in a single image. In the work of [5], they use unlabeled video sequences to improve the image semantic segmentation or instance segmentation of urban scenes segmentation. Instead, our method concentrates on improving the performance of weakly-supervised VSS methods with clicks in more complex scenes [8].

### 2.3 Knowledge Distillation

Besides model pruning [14, 17, 36], weight quantization [14, 44] and compact network design [22, 48], knowledge distillation [20] is first proposed as a novel technique for network compression. It is characterized by a Teacher-Student learning paradigm that propagates “dark knowledge” of a cumbersome teacher net to supervise the training of a tiny student net. Specifically, knowledge includes soft labels [20], intermediate features [19, 26, 45, 61], correlation information [33, 42, 43, 51], etc.

Although researches on knowledge distillation mostly focus on the image classification task, it can also be extended to more complex visual tasks, including object detection [4, 29], pose estimation [63] and image restoration [18, 28], which reveals the generality of this learning framework. The pioneer knowledge distillation work in semantic segmentation [57] transfers pixel-wise class probabilities and segmentation boundaries to the student net. Liu et al. [34] proposes structured distillation, including pair-wise distillation that extracts feature similarity in a local patch, and holistic distillation which captures higher-order semantics. Previous work in VSS that is related to our work is [35], which preserves temporal consistency by encoding motion information in distillation loss terms. In contrast, we concentrate on weakly-supervised learning with click annotations in VSS.

Self-distillation, also named teacher-free distillation, is proposed to prevent the usage of large teacher nets, which distills knowledge within the network itself. The commonly-used self-distillation strategy includes transferring high-level features to low-level features [21, 64], using soft labels of previous epochs to guide the training of the target epoch [60], etc. In this work, we design a temporal self-distillation mechanism, i.e. memory flow knowledge distillation training scheme, to boost the segmentation accuracy.

### 3 METHOD

In this section, we first introduce the preliminary weakly-supervised learning with click annotations in Section 3.1. Then we present the proposed WeClick pipeline, including the training and inference phases in Section 3.2. Finally we elaborate on the details of the overall training schemes in Section 3.3.

#### 3.1 Preliminary Weakly-Supervised Learning with Click Annotations

Training an image semantic segmentation network requires pairs of input image $x$ and its annotation $y$, where each pixel is assigned from $C$ classes. Let us denote $N$ as the total number of pixels in an image, $\hat{y}_{i}(c)$ as the $c^{th}$ class probability for pixel $x_i$ in the prediction $\hat{y} = F(x)$, where $F$ is a segmentation network. In the click-based
learning, only a few pixels of the training images are annotated, forming a partial map $m$ of the same size as $y$, where the element $m_i \in \{0, 1\}$. The sparsity of $m$ depends on whether the corresponding pixel is a click annotation. Since the unannotated pixels provide little information in the learning process, they are left out of consideration during the training phase. In general, partial Cross-Entropy (pCE) loss is used in the click-based learning algorithm:

$$L_{pCE}(\hat{y}, y) = \frac{\sum_{i=1}^{N} \sum_{c=1}^{C} m_i \cdot [-y_i(c) \log \hat{y}_i(c)]}{\sum_{i=1}^{N} m_i}.$$  (1)

In click-based learning, the point annotation is usually applied as the supervision signal seed and then propagated to the surrounding pixels through regularization mechanisms [40, 50] for obtaining more semantic information. In other words, the preliminary click-based learning algorithm consists of two parts, i.e. the pCE loss term and the regularization term. In this work, the Gated CRF [40] is used as the regularization term for its great ability in mining semantic information from click annotations.

### 3.2 WeClick Pipeline

We propose the WeClick pipeline for alleviating the fine-annotation scarcity issue in video semantic segmentation with effective training schemes. An overview of WeClick shows in Figure 3.

In the teacher net training phase, we pre-trained the large teacher net in the weakly training scheme with only click labels to obtain a well-performed segmentation network (see Figure 3(a)). As Figure 3(b) illustrates, during the student net training phase, we apply the pre-trained teacher net and the pre-trained flow net and freeze their parameters to assist the optimization of the student net. The input of the student net is based on multi-frames input pattern, which consists of the target frame $x_k$ with click annotations and the $n-1$ neighboring frames $\{x_{f_k}^{k-1}_{f_k=n+1}\}$ without any labels, where $f$ denotes the frame index. We train the target frame $x_k$ is fed to the student net to learn semantic information from click annotations; 2) Memory Flow knowledge Distillation (MFD) scheme which utilizes memory flow by distilling the neighboring predictions as extra supervised signals to the target frame via estimated optical flow; 3) Knowledge Distillation (KD) scheme, which learns soft knowledge from well-performed teacher net in both the target frame $x_k$ and the neighboring frames $\{x_{f_k}^{k-1}_{f_k=n+1}\}$. In the inference process, the teacher net and the optical flow net are removed and only the compact student net remains for single-frame inference with no additional computation cost.

### 3.3 Proposed Training Schemes

In this section, we elaborate on the detailed training schemes in WeClick, which consists of three parts: weakly training scheme, MFD strategy, and KD strategy.

**Weakly Training Scheme.** In WeClick, the only target frame $x_k$ has click annotation $y_k$, while other neighboring frames are unannotated. We applied weakly loss to the target frame prediction of student net $\hat{y}_k$, which is formulated as Eq. (2):

$$L_{weakly}(x_k) = L_{pCE}(\hat{y}_k, y_k) + \lambda L_{GCRF}(\hat{y}_k),$$  (2)

where GCRF stands for Gated CRF regularization terms [40] and $\lambda$ is the loss weight hyper-parameter.

**MFD Scheme.** MFD is a temporal self-distillation mechanism that distills knowledge within the network itself by transferring information from neighboring frames to the target frame via the estimated optical flow. The detailed MFD training scheme is illustrated in Figure 4(a). We apply a pre-trained motion estimation (i.e. optical

---

**Figure 3:** An overview of the proposed WeClick pipeline. (a) In the teacher net training phase, we pre-trained the cumbersome teacher net only in the weakly training scheme via click annotations to obtain a segmentation network with high accuracy. (b) During the student net training period, the teacher net and the flow net are pre-trained and their weights are fixed to assist the optimization of the student net. The input of the student net is based on multi-frames input pattern, which consists of the target frame $x_k$ (marked with red border in the Figure) with click annotations and the $n-1$ neighboring frames $\{x_{f_k}^{k-1}_{f_k=n+1}\}$ without any labels, where $f$ denotes the frame index. The overall optimization of student net consists of three training schemes: 1) weakly training scheme to realize weakly-supervised learning with click annotations in VSS; 2) Memory Flow knowledge Distillation (MFD) scheme that exploits memory flow in abundant unlabeled video data; 3) Knowledge Distillation (KD) scheme for model compression. (c) In the student net inference phase, the teacher net and the flow net are removed and only the compact student net remains for single-frame inference without additional computation cost.
net training phase, the consistency matrix adaptively decreases the weight of the unreliable region in the warped mask.

Neighboring frames is effectively propagated to the target frame, MFD training strategy, the memory flow from abundant unlabeled

objective for MFD training scheme is formulated as Eq. (3):

$$
\hat{L}_{MFD}(x_f, x_k) = W_f \odot ||\hat{y}_{f \rightarrow k}^S - \hat{y}_{f \rightarrow k}^\alpha||_2^2
$$

where $\odot$ denotes element-wise multiplication. With the proposed MFD training strategy, the memory flow from abundant unlabeled neighboring frames is effectively propagated to the target frame, which stimulates the optimization of the VSS model.

KD Scheme. To alleviate the inference latency issue, we adopt the knowledge distillation strategy to train the compact student net. Note that the teacher net is pre-trained with click annotations under the weakly training scheme in advance (see Figure 3(a)). We train the compact student net with soft knowledge from the teacher net using both the labeled frames and the abundant unlabeled frames. The goal of the student model is to align the class probability of each pixel in every corresponding mask generated by the teacher net. Based on the single-frame input $x_f$, the loss for KD is as follow:

$$
\mathcal{L}_{KD}(x_f) = KL(\hat{y}_f^S || \hat{y}_f^\alpha),
$$

where $\hat{y}_f^S$ and $\hat{y}_f^\alpha$ are the $f^{th}$ class probability maps of the student net and the teacher net, $C$ is the number of classes, $H$ and $W$ are the height and width of the prediction maps, and KL denotes Kullback-Leibler divergence.

Overall Optimization Process. The overall optimization process for student net consists of the above three training schemes. Note that only the target frame $x_k$ has click annotation and the $n-1$ neighboring frames $\{x_f \}_{f=k-n}^{k-1}$ remain unlabeled. The optimization goal for the training phase is to minimize Eq. (5):

$$
\mathcal{L} = \alpha \mathcal{L}_{weak}|y(x_k)| + \beta \sum_{f=k-n+1}^{k-1} \mathcal{L}_{MFD}(x_f, x_k) + \gamma \sum_{f=k-n+1}^{k-1} \mathcal{L}_{KD}(x_f),
$$

where the $\alpha$, $\beta$, and $\gamma$ are modulating factors for the loss weights.

4 EXPERIMENTS

4.1 Datasets and Metrics

Datasets. We conduct the experiments on the standard benchmarks, i.e. Cityscapes [8] and Camvid [2], for video semantic segmentation (VSS). Cityscapes consists of 5000 sparsely-labeled video snippets of urban scenes, each of which contains 30 frames and only the
20th frame is finely annotated in pixel level. The images are divided into 2975, 500, 1525 for training, validation, and testing respectively. Only 19 semantic classes and 1 void class are used for training. As for the Camvid, it contains 4 videos that are annotated at 1 Hz, with 367 images for training, 100 for validation, and 233 for testing.

**Metrics.** We apply mean Intersection-over-Union (mIoU) [37] and mean Pixel Accuracy (mPA) as the accuracy metrics. Besides, we report the parameters (#Params) and Frames Per Second (FPS) to evaluate the model efficiency.

### 4.2 Implementation Details

**Click Annotations.** In our experiments, we only use click annotations instead of per-pixel annotations for training. Referring to the click generation strategy in [1] and [40], we first divide the classes into instance classes (e.g. car and person), and non-instance classes (e.g. building, and road). Concerning instance classes, instance segmentation masks are used to distinguish different instances. In terms of non-instance classes, 8-connectivity component labeling is applied and components with less than 512 pixels are discarded. For the remaining non-instance objects and all the instance objects, we select the centroid point of the semantic object as the click annotation if the point is within the object. Otherwise, we randomly sample one point within the object as the click annotation. It is noteworthy that the click annotation can be an arbitrary point within the semantic object in practice.

**Network Architectures.** We apply two popular semantic segmentation networks, i.e. DeeplabV3+ [6] and PSPNet [65] to perform the experiments. For the teacher net, ResNet101 [16] is used as backbone. With respect to compact student net, we adopt ResNet18 and MobileNetV2 [48]. Besides, We employ FlowNetV2 [24] to estimate the optical flow between two frames.

**Experimental Setup.** The experiments are performed under a single Nvidia Tesla V100 GPU, and Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz. During the training stage, common practices are applied, i.e. backbone pre-training with ImageNet [10]. Xavier initialization [13], data augmentations including random crop, random scale in [0.5, 2.0], random horizontal flip, random mirror, etc. As for the Cityscapes dataset, we train the model using stochastic gradient descent (SGD) with initial learning rate $7e^{-3}$, polynomial learning rate decay scheduler power 0.9, momentum 0.9, weight decay $1e^{-4}$, batch size 4, and crop size 768 $\times$ 768 for 120 epochs. For the Camvid dataset, we conduct experiments with initial learning rate $1e^{-4}$, batch size 8, and crop size 360 $\times$ 360 for 200 epochs. Besides, WeClick introduces several hyper-parameters, including memory flow direction, input frame quantity, and sampling policy. ‘Memory flow direction’, is the direction of memory flow conveyed within the semantic object in practice.

### 4.3 Ablation Studies

**Effectiveness of the Proposed Training Schemes.** To validate the effectiveness of the training schemes, we conduct extensive experiments on DeeplabV3+ with ResNet18 and MobileNetV2. Due to the great regularization capability of the proposed WeClick pipeline, the model accuracy increases dramatically. As Table 1 shows, the mIoU gains of ResNet18 and MobileNetV2 reach 5.42% and 5.43% respectively. It is noteworthy that the compact student net with ResNet18 achieves 58.30% in terms of mIoU on Cityscapes validation set, nearly surpassing the teacher net with ResNet101, which indicates the proposed method significantly boosts the performance of the lightweight networks in the click-based VSS.

**Impact of the MFD Direction.** In WeClick, MFD propagates memory flow from preceding frames to the target frame by default, which is denoted as forward MFD. We extend the direction of MFD to three cases, including forward, backward, and bi-direction. In this ablation study, we set the input frame quantity $n$ to 3 (i.e. including 1 target frame and 2 neighboring frames) and select the neighboring frames with the fixed frame-interval 1. Note that we consider the 20th frame in each video snippet as the target frame in Cityscapes.
Therefore, backward MFD indicates information transfers from subsequent frames (i.e. 21st and 22nd frames) to the target frame, while bi-direction MFD conveys information from both preceding and subsequent frames (i.e. 19th and 21st frames) to the target frame. Table 2 demonstrates that forward MFD outperforms the other two cases in terms of mIoU on student nets DeeplabV3+ with ResNet18 and MobileNetV2 as backbones.

Impact of the Input Frame Quantity. In this section, we investigate the impact of input frame quantity $n$ (i.e. including 1 target frame and $n - 1$ neighboring frames), in student net DeeplabV3+ with ResNet18 under both forward and bi-direction MFD settings with the fixed frame-interval 1. Note that we conduct a single-frame experiment without MFD to act as the baseline. The model trained with 3 input frames performs best.

| Sampling Policy         | Forward mIoU (%) | mPA (%) | Bi-Direction mIoU (%) | mPA (%) |
|-------------------------|------------------|---------|-----------------------|---------|
| Fixed Frame-Interval 1  | 58.03            | 65.99   | 57.56                 | 66.16   |
| Fixed Frame-Interval 4  | 57.94            | 66.03   | 57.43                 | 66.09   |
| Fixed Frame-Interval 7  | 57.83            | 65.50   | 57.41                 | 65.69   |
| Random Sampling         | **58.30**        | **66.39** | **57.74**             | **66.43** |

Table 3: Comparison on different sampling policies in the student net DeeplabV3+ with ResNet18 on 3 input frames (including the target frame). Larger sampling intervals are not conducive to improving performance, while random sampling policy achieves performance gain.

Figure 5: Comparison on different input frame quantities in student net DeeplabV3+ with ResNet18 under forward and bi-direction MFD settings. Note that we conduct a single-frame experiment without MFD to act as the baseline. The model trained with 3 input frames performs best.

Table 4: Accuracy (mIoU/mPA) and inference speed (FPS) on Cityscapes validation set. For short notation, DL-R101, DL-R18, and DL-Mob denote DeeplabV3+ with ResNet101, ResNet18, and MobileNetV2 as backbones. PSP-R101, PSP-R18, and PSP-Mob represent PSPNet with ResNet101, ResNet18, and MobileNetV2 as backbones, respectively. Our proposed WeClick improves both the accuracy and inference speed substantially than baseline.

| Student Teacher | mIoU (%) | mPA (%) | #Params (M) | FPS |
|-----------------|----------|---------|-------------|-----|
| DL-R101 None    | 58.35    | 67.24   | 59.34       | 8.41|
| DL-R18 None     | 52.88    | 62.21   | 16.61       | 22.06|
| DL-R18 DL-R101  | **58.30**| **66.39**| 16.61       | 22.06|
| DL-Mob None     | 51.99    | 62.77   | 13.35       | 25.97|
| DL-Mob DL-R101  | **57.42**| **66.89**| 13.35       | 25.97|

Table 4: Accuracy (mIoU/mPA) and inference speed (FPS) on Cityscapes validation set. For short notation, DL-R101, DL-R18, and DL-Mob denote DeeplabV3+ with ResNet101, ResNet18, and MobileNetV2 as backbones. PSP-R101, PSP-R18, and PSP-Mob represent PSPNet with ResNet101, ResNet18, and MobileNetV2 as backbones, respectively. Our proposed WeClick improves both the accuracy and inference speed substantially than baseline.

In this section, we investigate the impact of different sampling policies on student net with 3 input frames, including the fixed frame-intervals (i.e. 1, 4, and 7), and random sampling strategy in a predefined range (i.e. from 16th to 19th in the preceding frames and from 21st to 24th in the subsequent frames). For instance, in the setting of forward MFD with frame-interval 4, the selected input frames are 12th, 16th and 20th. Table 3 reveals that, on the one hand, with a larger fixed sampling frame-interval, the performance declines because of the optical flow estimation noise between two distant frames. On the other hand, with the random sampling policy, the mIoU is improved slightly compared to the best results in the fixed frame-interval settings, i.e. frame-interval 1. It can be explained that random sampling policy sees more unlabeled video data, and capture both short-term and long-term temporal information when training.

Impact of the Teacher Net and Student Net. In this section, we show the generalization of our WeClick in different teacher-student settings. Table 4 represents that our proposed method consistently improves the performance of student nets without compromising efficiency during inference. Specifically, student net PSPNet with ResNet18 surpasses teacher net and outperforms baseline by 4.72% mIoU. Moreover, MobileNetV2 in both DeeplabV2 and PSPNet reach real-time execution under Nvidia Tesla V100 with only slight accuracy compromise compared with the teacher net, which indicates the effectiveness and generalization of WeClick.
Figure 6: Comparison between WeClick and baseline in per-class IoU on DeeplabV3+ with ResNet18. WeClick improves the performance for instance class, such as truck, bus, train, etc, which benefits from the memory flow introduced by MFD.

### Table 5: Comparison with weakly-supervised learning VSS methods on Cityscapes and Camvid.

| Method                          | Task   | Supervision | #Params(M) | Cityscapes mIoU (%) | FPS | Camvid mIoU (%) | FPS |
|---------------------------------|--------|-------------|-----------|---------------------|-----|----------------|-----|
| Bringing [46]                   | VSS    | image-level | –         | 24.9                | -   | –              | -   |
| ADVENT [52]                     | ISS    | UDA         | –         | 48.0                | 7.1 | –              | –   |
| Gated CRF (DeeplabV3+ with ResNet34) [40] | ISS    | click-level | 26.72     | 57.43               | 17.3 | 46.9           | 99.1 |
| Gated CRF (DeeplabV3+ with ResNet18) [40] | ISS    | click-level | 16.61     | 52.88               | 22.1 | 46.2           | 128.3 |
| Our WeClick (DeeplabV3+ with ResNet18) | VSS    | click-level | 16.61     | 58.30               | 22.1 | 48.7           | 128.3 |
| Our WeClick (DeeplabV3+ with MobileNetV2) | VSS    | click-level | 13.35     | 57.42               | 26.0 | –              | –   |
| Our WeClick (PSPNet with ResNet18) | VSS    | click-level | 13.80     | 55.25               | 15.6 | –              | –   |
| Our WeClick (PSPNet with MobileNetV2) | VSS    | click-level | 10.31     | 52.44               | 52.2 | –              | –   |

Table 5 demonstrates that our method outperforms other state-of-the-art weakly-supervised methods, including image-level, click-level, and unsupervised domain adaptation (UDA). The results show WeClick (DeeplabV3+ with ResNet18) outperforms other methods consistently by a large margin and WeClick with MobileNetV2 reaches a real-time inference speed.

### 4.4 Comparison with the State-of-the-art Methods

In this section, we compare our proposed WeClick with state-of-the-art weakly-supervised methods, whose results are from their papers. Table 5 demonstrates that our method outperforms other state-of-the-art weakly-supervised methods, including image-level, click-level and UDA weakly-supervised algorithms, by a large margin. For example, WeClick (DeeplabV3+ with ResNet18) improves the segmentation performance than [46] and [52] by 33.4% mIoU and 10.3% mIoU, respectively. It is noteworthy that WeClick is superior to state-of-the-art click-based image semantic segmentation method Gated CRF [40] in both accuracy and inference speed.

Besides, Figure 6 shows the per-class IoU of the methods with and without WeClick. Our method significantly improves the performance of instance classes, such as truck, bus, train, etc. It can be explained that the memory flow enables the object classes with significantly changing motion to perform better.

We further conduct additional experiments in DeeplabV3+ with ResNet18 on the CamVid dataset to validate the generalization of the WeClick across different datasets. As Table 5 shows, our proposed WeClick stimulates the capability of the model and outperforms the baseline by 2.5% mIoU. Our method can achieve 128.3 fps with a 352 x 480 resolution on a single Nvidia Tesla V100. Consistent mIoU improvements on both the Cityscapes and the Camvid verify both the generalization and effectiveness of our proposed WeClick pipeline for click-based VSS.

### 5 CONCLUSIONS

In this work, we make the first attempt to propose an effective weakly-supervised video semantic segmentation pipeline with click annotations, called WeClick, to save tedious annotating effort by segmenting an instance of the semantic class with only a single click during the training phase. To boost the performance, we design a novel memory flow knowledge distillation training scheme, which utilizes temporal information in the unlabeled video frames. WeClick learns compact video semantic segmentation models with the low-cost click annotations in the training period yet achieves real-time and accurate models during the inference phase. Extensive experiments on the Cityscapes and Camvid show that the compact models with the proposed training schemes outperform state-of-the-art methods by a large margin, which demonstrates the effectiveness and generalization of WeClick. We hope our work could provide insights for researchers of this field to design novel schemes for weakly-supervised video semantic segmentation.
[57] Jiafeng Xie, Bing Shuai, Jian-Fang Hu, Jingyang Lin, and Wei-Shi Zheng. 2018. Improving fast segmentation with teacher-student learning. In BMVC.
[58] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. 2018. Dynamic video segmentation network. In CVPR. 6556–6565.
[59] Xiyu Yan, Huahui Gong, Yong Jiang, Shu-Tao Xia, Feng Zheng, Xinge You, and Ling Shao. 2020. Video scene parsing: An overview of deep learning methods and datasets. Computer Vision and Image Understanding 201 (2020), 103077.
[60] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L. Yuille. 2019. Snapshot distillation: Teacher-student optimization in one generation. In CVPR. 2859–2868.
[61] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In CVPR. 4133–4141.
[62] Sergey Zagoruyko and Nikos Komodakis. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016).
[63] Feng Zhang, Xiatai Zhu, and Mao Ye. 2019. Fast human pose estimation. In CVPR. 3517–3526.
[64] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In ICCV. 3713–3722.
[65] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid Scene Parsing Network. In CVPR. 2881–2890.
[66] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Deep feature flow for video recognition. In CVPR. 2349–2358.