A. Related Work

Video Semantic Auto-Annotation. Pseudo-labeling and semi-supervised learning are the two popular types of methods for automating video semantic segmentation annotations. The pseudo-labeling approaches [27] use a pre-trained teacher model to generate labels for the test video sequences. However, these approaches are typically frame-based and do not consider the rich temporal constraints in the videos. Therefore, the pseudo-labels are inevitably noisy, especially when the pre-trained model is trained with the data from a different domain from the input video. This work proposes a novel test-time fine-tuning method to adapt the pre-trained model to the specific video to generate pseudo labels more accurately.

Among the semi-supervised learning approaches, Label Propagation (LP) is widely adapted [3, 2, 30, 18, 32]. Most work uses optical flow to guide the LP process. These methods rely on accurate optical flow estimation, which is difficult to obtain. Otherwise, the erroneous flow estimation can result in propagated labels that are misaligned with their corresponding frames. Rather than conducting direct LP, our method uses the optical flow to generate consistency constraints as a loss to fine-tune the segmentation model, which makes it more robust to flow noise. Moreover, the proposed fine-tuning considers both semantic and temporal information to predict temporally consistent semantic annotations across the full video without the limitations of traditional LP methods.

Active Learning. Rooted in traditional machine learning, active learning [39] allows learners to *actively* query the specific labels they want to obtain, saving labeling costs dramatically. Inspired by the success of active learning, previous methods [17, 40] studied how to select instances to refine a network for segmentation tasks. Our framework's objective is different from them, as we are querying samples from a video such that their annotations could boost the label propagation accuracy on the input video. There is one work [43] studies the active frame selection problem for label propagation. Our work is different in two ways: First, the method in [43] selects frames for just once, while our method could select video frames, rectangles of frames, or even super-pixels in a human-in-the-loop manner for multiple iterations. Second, the existing method closely ties with a particular LP technique and does not comply with modern deep networks. Our method is generic and can work with different segmentation networks.

Human-in-the-Loop for Visual Annotations. Some researchers aim to reduce the annotation cost in a human-in-the-loop model learning [1, 34]. And [20, 12] studied the interactive video object segmentation frameworks. However, solving video semantic segmentation problem in the human-in-the-loop framework has never been studied.

B. Detailed Experimental Settings

B.1. Training Settings

We perform three iterations of ASS for each testing sequence. The annotation budget for each iteration is divided equally from the total budget.

We use the HRNet-W48 [44] as the backbone network (other networks can be easily incorporated). We set the consistency loss weight $\lambda = 1$. The initial learning rate in each iteration is 0.004. In each iteration, we fine-tune the network for 15 epochs with a learning rate of 0.004 and SGD optimizer [37] with momentum 0.9. We follow the "poly" learning rate policy to reduce the learning rate gradually. The batch size is 14 for SceneNet RGB-D [29] dataset, and 2 for VEIS [38] dataset.

In the ASS diversity sampling, we transform each frame to feature space using the previous iteration backbone network without segmentation head:

$$F_t = \psi_{\theta_{k-1}}(I_t), \tag{7}$$

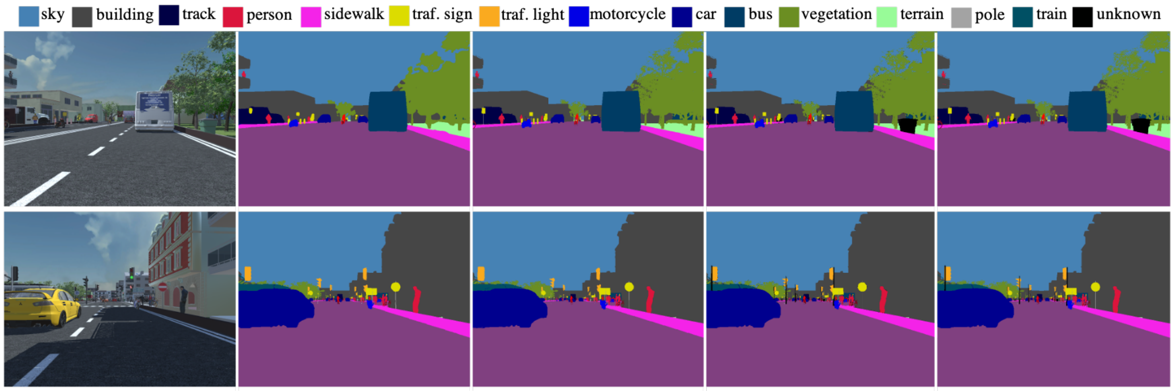
where ψ denotes the segmentation network backbone. More concretely, the feature map F_t is the concatenation of four feature maps after the fourth stage of HRNet-W48. The number of channels of F_t is 720.

We use the RAFT-things checkpoint [42] to generate the flow correspondence and use SpixelFCN-bsd checkpoint [21] to generate the super-pixels. All the settings in RAFT and SpixelFCN are as default. We use openCV [5] connectedComponents, findContours, and approxPolyDP functions to estimate the "Boundary Click" positions. The parameter ϵ in approxPolyDP controlling the fineness of simplified polygons is set to 1.

For HRNet backbone, RAFT and SpixelFCN, we use the code from their official implementation in Pytorch [33]. All the experiments run on machines with 4$\times$Nvidia 1080s.

B.2. Evaluation and Metrics.

We use four metrics to evaluate our method thoroughly, which are pixel accuracy, mean Intersection over Union (mIoU), boundary Intersection over Union (Boundary-IoU), and temporal consistency. The first two are commonly used in segmentation tasks to measure the accuracy of predictions. Here we only illustrate the last two metrics in detail.



(a) Raw (b) 85% mIoU (c) 90% mIoU (d) MA (e) GT

Figure 10. Visualization of our generated annotations in VEIS. (a) is the video frame, (b) costs about 10.4% clicks with annotating Frame, and (c) costs about 44% clicks with annotating Frame. (d) is the mimic manual annotation, and (e) is the ground-truth.

Boundary-IoU. The boundary-IoU is proposed in [11]. The boundary-IoU between our prediction $\mathbf{Q}$ and ground-truth $\mathbf{G}$ is calculated as:

$$\text{boundary-IoU}\left(\mathbf{G},\mathbf{Q}\right) = \frac{\left|\left(\mathbf{G}_d \cap \mathbf{G}\right) \cap \left(\mathbf{Q}_d \cap \mathbf{Q}\right)\right|}{\left|\left(\mathbf{G}_d \cap \mathbf{G}\right) \cup \left(\mathbf{Q}_d \cap \mathbf{Q}\right)\right|}, \tag{8}$$

where $\mathbf{G}_d$ and $\mathbf{Q}_d$ are the sets of pixels in the boundary region of the ground-truth mask and the prediction mask respectively. d is the pixel width of the boundary region. We set d to 2 in all experiments.

Temporal consistency (TC). We also measure the temporal consistency (TC) of the generated annotations by measuring the mIoU between two consecutive predictions similar to [26].

We measure the temporal consistency (TC) of the generated annotations by measuring the mIoU between two consecutive predictions similar to [26]. The TC between frame t and frame $t-1$:

$$TC\left(\mathbf{Q}_{t-1},\mathbf{Q}_t\right) = \frac{\left|\mathbf{Q}_t \cap \hat{\mathbf{Q}}_{t-1}^{(M)}\right|}{\left|\mathbf{Q}_t \cup \hat{\mathbf{Q}}_{t-1}^{(M)}\right|}, \tag{9}$$

where $\mathbf{Q}_t$ is the prediction of frame t, $\mathbf{Q}_{t-1}$ is the prediction of frame $t-1$. $\hat{\mathbf{Q}}_{t-1}^{(M)}$ is the warped prediction from frame $t-1$ to frame t, and pixels where does not pass the forward-backward check $M_{t-1\rightarrow t}$ will be marked as ignored label in $\hat{\mathbf{Q}}_{t-1}^{(M)}$. The calculation of $TC\left(\mathbf{Q}_{t-1},\mathbf{Q}_t\right)$ is very similar with calculating the standard IoU in the segmentation task, where we treat prediction as $\mathbf{Q}_t$, and ground truth as $\hat{\mathbf{Q}}_{t-1}^{(M)}$. So the TC on all the test sequences can be calculated similar to IoU and mIoU on the whole test set in a segmentation task.

C. More Experimental Results on VEIS Dataset

Table 3. This table shows the most efficient sample granularity for different mIoU benchmarks in VEIS. The last row represents manually annotating all the frames.

Annotation mIoU	Granularity	Anno. Clicks	Anno. Pixel
80%	Rec40	4.3%	1.3%
85%	Frame	10.4%	10%
90%	Frame	44%	51%
96%	Frame	100%	100%

Click Cost for Label Quality Benchmarks. Tab. 3 lists the least annotation clicks required to generate 80%, 85%, and 90% mIoU labels, and the corresponding sample granularity. Notice that in the VEIS dataset, the mIoU for fully manual annotation is 96%, which is lower than 99% of SceneNet. This difference is mainly because the RDP algorithm simplifies the over-detailed “bicycle” ground-truth in VEIS. As a result, we test up to 90% mIoU annotation quality. Annotating Rec40 makes the model generate 80% mIoU annotations with the least clicks, which is only 4.3%. Annotating frames outperforms other granularity settings when targeting 85% and 90% mIoU annotations. The percentages of clicks are 10.4% and 44% respectively.

We noticed that the percentage of clicks needed in VEIS is much larger than in SceneNet. This is mainly because VEIS images are of higher resolution, and objects include more details. This leads to more clicks in segmentation annotation.

Comparison of Boundary-IoU. In VEIS, the observation is similar to it in SceneNet: Larger granularity samples give better predictions on the boundary, and training on SP gives the worst boundary quality.

Generated annotation visualization. Fig. 10 shows the visualizations in VEIS. The 85% mIoU annotations in (b) cost 10.4% clicks, and only details of “tree” are imperfect. The 90% mIoU annotations in (c) cost 44% clicks and are without a significant visual difference to MA.