Less is More: Surgical Phase Recognition from Timestamp Supervision

Xinpeng Ding, Xinjian Yan, Zixun Wang, Wei Zhao, Jian Zhuang,
Xiaowei Xu and Xiaomeng Li, *Member, IEEE*

***Abstract*— Surgical phase recognition is a fundamental task in computer-assisted surgery systems. Most existing works are under the supervision of expensive and time-consuming full annotations, which require the surgeons to repeat watching videos to find the precise start and end time for a surgical phase. In this paper, we introduce timestamp supervision for surgical phase recognition to train the models with timestamp annotations, where the surgeons are asked to identify only a single timestamp within the temporal boundary of a phase. This annotation can significantly reduce the manual annotation cost compared to the full annotations. To make full use of such timestamp supervisions, we propose a novel method called uncertainty-aware temporal diffusion (UATD) to generate trustworthy pseudo labels for training. Our proposed UATD is motivated by the property of surgical videos, *i.e.*, the phases are long events consisting of consecutive frames. To be specific, UATD diffuses the single labelled timestamp to its corresponding high confident (*i.e.*, low uncertainty) neighbour frames in an iterative way. Our study uncovers unique insights of surgical phase recognition with timestamp supervision: 1) timestamp annotation can reduce 74% annotation time compared with the full annotation, and surgeons tend to annotate those timestamps near the middle of phases; 2) extensive experiments demonstrate that our method can achieve competitive results compared with full supervision methods, while reducing manual annotation costs; 3) less is more in surgical phase recognition, *i.e.*, less but discriminative pseudo labels outperform full but containing ambiguous frames; 4) the proposed UATD can be used as a plug-and-play method to clean ambiguous labels near boundaries between phases, and improve the performance of the current surgical phase recognition methods. Code and annotations obtained from surgeons are available at `https://github.com/xmed-lab/TimeStamp-Surgical`.**

***Index Terms*— Surgical phase recognition, timestamp supervision, uncertainty estimation**

I. Introduction

COMPUTER-ASSISTED surgery systems can improve the surgery's quality and ensure the patients' safety in modern operating rooms [1], [2]. Surgical phase recognition is one key component of computer-assisted surgery systems, which aims to predict which phase is occurring at the current frame [3], [4]. It can be used for automatic indexing of surgical video databases [5], monitoring surgical process [6], scheduling surgeons [7] and assessing surgeons' skills [8]. In recent years, automated surgical phase recognition has featured deep learning [9]–[11] and has reached promising recognition performance [5], [12], [13]. Most current surgical phase recognition approaches require full annotations from surgeons, *i.e.*, the surgeons need to find the precise start and end time for a surgical phase. To this end, the surgeon should repeat watching the video at a very slow speed to find a specific time for the start of the phase. Then, the surgeon needs to continue to watch the video and find the precise end time of the phase. As shown in Fig. I (a), this full annotation is very time-consuming, *e.g.*, surgeons need to spend an average of 562.83 seconds to annotate a video. Furthermore, the boundaries between different phases are usually ambiguous [12]. Due to the subjective of different surgeons, they would provide inconsistent annotations for the same video [14].

To address the limitation of the full annotation, this paper introduces the ***timestamp supervision*** to surgical phase recognition which trains the model from the timestamp annotation as

Manuscript received XX XX, 2021. This work was supported in part by the research grant from Shenzhen Municipal Central Government Guides Local Science and Technology Development Special Funded Projects under Grant 2021Szvup139, in part by a research grant from HKUST-BICI Exploratory Fund (HCIC-004) and in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Reference Number: T45-401/22-N)

X. Ding is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, SAR, China (e-mail: xdingaf@connect.ust.hk).

X. Yan is with the Department of Cardiovascular Surgery, Guangdong Cardiovascular Institute, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China; and also with the Department of Cardiovascular Surgery, Guangdong Provincial People's Hospital (Guangdong Academy of Medical Sciences), Southern Medical University, Guangzhou, China. (email: yanxinjian@gdph.org.cn).

Z. Wang is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, SAR, China (e-mail: craddywang@gmail.com).

W. Zhao is with the School of Physics, Beihang University, Beijing, China, and also with Beihang Hangzhou Innovation Institute Yuhang Xixi Octagon City, Yuhang District, Hangzhou, China (email: zhaow20@buaa.edu.cn).

J. Zhuang and X. Xu are with the department of Cardiovascular Surgery, Guangdong Provincial People's Hospital (Guangdong Academy of Medical Sciences), Southern Medical University, Guangzhou, China, and also with the Department of Cardiovascular Surgery, Guangdong Cardiovascular Institute, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou, China (email: zhuangjian@gdph.org.cn and xiao.wei.xu@foxmail.com).

X. Li is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, SAR, China, and also with The Hong Kong University of Science and Technology Shenzhen Research Institute, Shenzhen 518057, China (e-mail: eexmli@ust.hk).

X. Ding and X. Yan contributed equally. J. Zhuang, X. Xu and X. Li are the corresponding authors.

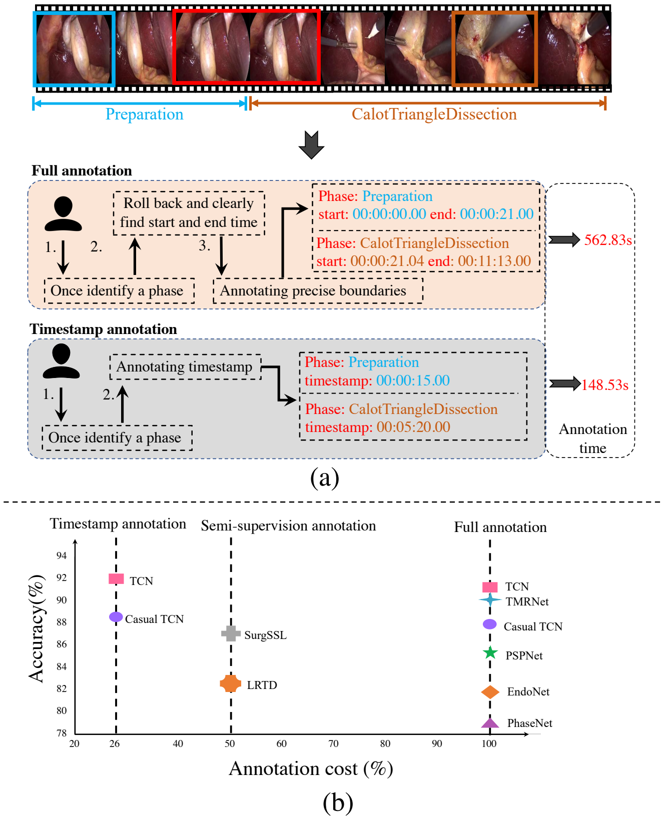


Fig. 1. (a) Comparison of the full annotation and our proposed timestamp annotation. When labelling a phase in full annotation, the annotator needs to roll back and find the precise start and end time. In our timestamp annotation, only a single timestamp is labelled without identifying the start and end time, which can save annotation cost and is much faster than the full annotation. We invite two surgeons to conduct full and timestamp annotations, and record their annotation times. We finally observe that they took an average of $\mathbf{562.83}$ and $\mathbf{148.53}$ seconds per video for full annotation and timestamp annotation, respectively. (b) The trade-off between manual annotation cost and accuracy for different methods. Compared with existing methods, our method achieves the competitive performance while using only $\mathbf{26\%}$ manual annotation cost compared with the full supervision.

shown in Fig. I (a). In timestamp annotation, the surgeons only annotate the phase class and a single timestamp for each phase, instead of start and end times. Once identifying the phase, the surgeon records the current timestamp (e.g., 00:05:20.00), no need to roll back and repeat watching the video to find a precise start time. After recording this single timestamp, since there is no need to find the end time, the surgeon would continue to go through the video quickly to find another phase. Hence, the timestamp annotation significantly reduces manual annotation cost compared to the full annotations; see the detailed annotation analysis in Section IV-B. Given timestamp supervision, *i.e.*, only a single label for each phase, the total number of positive frames is quite small, and the naive way that training with annotated labels may be difficult to learn a robust model; see results in Table. II. To generate more pseudo labels, some researchers propose to detect the action changes between two consecutive labeled frames for action recognition in natural videos with timestamp supervision [14]. However, this method displays limited performance to surgical videos because surgical videos contain more ambiguous boundaries, leading to the noisy and inconsistent pseudo labels; see Sec. IV-E for detailed discussion.

To address the above problems, we leverage the property of surgical videos to generate more trustworthy pseudo labels from timestamp supervision. The property we observed is that phases in the surgical video are long events consisting of continuous frames, which shows a desirable temporal property that the closer the frames to the annotated timestamp, the more likely they are to be classified to the same label as the annotated one. Frames far from the annotated timestamp are difficult to have correct pseudo labels. Based on the above property, a **U**ncertainty-**A**ware **T**emporal **D**iffusion (**UATD**) module is proposed to diffuse the annotated timestamps to their adjacent low-uncertainty frames in the temporal axis. In this way, only frames with high confidence and near the annotated timestamps would be considered for adding into pseudo-labels for training. Furthermore, the duration of the surgical videos generally last tens of minutes or even hours, making it hard to train the model in an end-to-end manner. Current works [3], [4], [15] generally sample a few consequent frames from the long videos, and optimize the combined spatial-temporal model in an end-to-end manner. This can be implemented in the full annotations, since all sampled consequent frames have labels. However, in timestamp annotation, most of the sampled frames have no labels, resulting in the imbalance of positive and negative samples. This imbalance training would degrade the performance; see details in Table III. To this end, we propose **L**oop **T**raining (**LP**), which optimizes the spatial and temporal model in an independent and iterative way.

We conduct empirical studies based on the proposed UATD and LP, and discover important insights of surgical phase recognition from timestamp supervision as follow: **1)** Timestamp annotation can reduce 74% annotation time compared with the full annotation, and surgeons tend to annotate those timestamps that are near the middle of phases; see details in Fig. 3. **2)** Extensive experiments demonstrate that our method can achieve competitive results compared with full supervision methods, while reducing manual annotation cost; see details in Table I. **3)** Less is more in surgical phase recognition, *i.e.*, less but discriminative pseudo labels outperform full but containing ambiguous frames; see details in -Table. I. **4)** The proposed UATD can be used as a plug-and-play method to clean ambiguous labels near boundaries between phases, and improve the performance of the current surgical phase recognition methods; see details in Fig 12. The reason is that training with our method would help to decrease intra-class distance and increase inter-class distance simultaneously; see details in Table. IX. The main contributions of this work can be summarized as the following:

- We study surgical phase recognition with a new timestamp supervision, which is the most efficient annotation setting in current surgical works. We invite two surgeons with rich clinical experience to annotate timestamp annotations and record their annotation time, and find that the timestamp annotation can reduce 74% annotation cost compared with the full annotation.

- We introduce UATD to generate the trustworthy pseudo labels from the timestamp annotation, and LP to train the model from the generated pseudo labels in an iterative way.
- We conduct in-depth empirical studies of the proposed UATD and LP based on timestamp supervision, and discover four deep insights which may boost the future development of surgical phase recognition.

II. Related Work

A. Surgical Phase Recognition

We broadly classified related methods for surgical phase recognition into two categories including fully-supervised learning and label-efficient learning.

Fully-supervised Learning. In fully-supervised learning, each frame in a surgical video is labeled. Early works [16]–[18] use hand-crafted features such as color and texture to perform recognition, which achieves limited performance and poor generalization. With the development of neural networks, recent deep learning-based methods achieve the great success [3]–[5], [12], [13], [15], [19]–[23]. ZIBNET [24] a state-preserving Long Short Term Memory (LSTM) to utilize the long-term evolution of tool usage within complete surgical phases. EndoNet [5] first uses a convolutional neural network to automatically learn features and prove its effectiveness for surgical phase recognition. SV-RCNet [3] integrates convolutional neural networks (CNN) and long short-term memory (LSTM) to learn both spatial and temporal representations in an end-to-end way. To capture the long-range temporal relationship, TMRNet [4] introduces a memory bank and TeCNO [25] uses dilated temporal convolutional network to get a large receptive field. Recently, Yi *et al.* [20] realize the negative effect of hard frames and propose data cleansing and online hard frames mapper to detect and handle them respectively. Yi *et al.* [21] find that simply applying multi-stage architecture *e.g.* multi-stage TCN makes the refinement fall short and thus design not end-to-end training manner to alleviate this problem. OperA [26] leverages attention weight to yield further insights into the decision-making process. Trans-SVNet [13] proposes a hybrid embedding aggregation Transformer to fuse spatial and temporal embedding. Ding *et al.* [12] emphasize the importance of segment-level semantics and extract semantic-consistent segments to refine the erroneous predictions. Notably, some related methods [5], [15], [25] utilize additional tool presence labels to perform a multi-task learning to facilitate surgical phase recognition.

Semi-supervised Learning. Despite the great success the above methods get, they require a large amount of annotated videos, which is very costly [27], [28]. some researchers [29]–[33] explore the methods for semi-supervision, where only parts of videos in the dataset are fully annotated, and others are unlabelled. For example, LRTD [32] use active learning to this context. It captures the long-range temporal dependency among continuous frames in the unlabeled data pool and selects the clips with weak dependencies to annotate. Yengera *et al.* [30] introduce self-supervised pre-training ensuring all available laparoscopic videos can be utilized. Yu *et al.* [29] propose a teacher/student approach where the teacher is trained on a small set of labeled videos and generates pseudo labels on the rest of unlabeled videos for student model learning. Furthermore, SurgSSL [33] uses consistency regularization and pseudo-labeling to leverage the knowledge in unlabeled data, which progressively leverages the inherent knowledge held in the unlabeled data to a larger extent.

Comparison of the manual annotation cost for different supervision setting. Here, we compare our proposed timestamp annotation compared with the above methods including full supervision methods and semi-supervision methods. In full supervised methods [12], [25], annotators are required to repeat watching the video and roll back to find the precise start and end time for each phase, which is very time-consuming. As shown in Fig. I (a), the average annotation time of each video for full supervision is 562.83s. In semi-supervision [29]–[33], the authors are required to only label full annotations for a few parts of all videos. Generally, in semi-supervised surgical phase recognition methods, 50% of videos are required to be annotated for achieving competitive results compared with full supervision methods, as shown in Fig. I (b). However, the annotation times of the introduced timestamp supervision is only 148.53s for each video, *i.e.*, 26% annotation time of the full supervision, and achieve the competitive results. For clarity, using the same network TCN [14], our methods achieve 91.9% accuracy in Cholec80 with only 26% annotation time, while the full supervision achieves 91.1% accuracy using 100% annotation time. Meanwhile, the SOTA semi-supervised method SurgSSL [33] achieves 87.0% accuracy using 30% annotation times. Hence, our proposed method is the best trade-off between accuracy and manual annotation cost.

B. Weak Supervision for Video Understanding

Weakly supervision has received widespread attention in some video understanding tasks, such as temporal action localization [34]–[40] and action segmentation [41]–[43]. Some of them use video-level supervision, *i.e.*, a set of action categories, while some use transcript-level supervision, *i.e.*, an ordered list of actions. For example, Richard *et al.* [44] leverage text-based grammar from unordered action sets. Although they significantly reduce the annotation effort, the performance is quite limited. To trade off the annotation-efficient and performance, timestamp supervision [14], [45]–[47] is proposed for action recognition. For example, SF-Net [47] designs an action frame mining and a background frame mining strategy to introduce more negative frames into the training process. However, the above methods aiming at temporal action localization task generate very limited pseudo labels, is not suitable for surgical phase recognition, *i.e.*, frame-wise recognition. In the action segmentation task, to generate frame-wise pseudo labels, Li *et al.* [14] detect the action change between two consecutive timestamps by stamp-to-stamp energy function and generate full pseudo labels. However, in surgical videos, the frames near boundaries are generally ambiguous, and the generated pseudo labels may be noisy annotations, which degrades the performance. Compared with previous approaches, our proposed method generates as

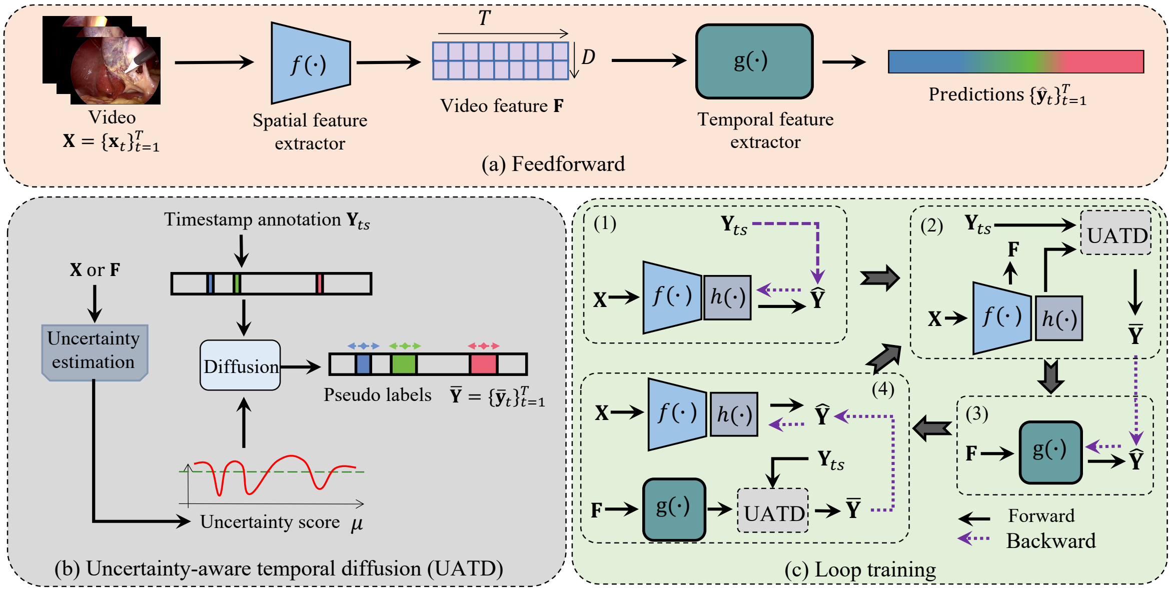


Fig. 2. Overview of our proposed framework. (a) The feedforward process of mapping a video to the phase predictions. A video is first fed into a spatial feature extractor (normally a CNN) to obtain the video feature, followed by a temporal feature extractor to obtain the frame-wise prediction. (b) Uncertainty-aware temporal diffusion (UATD). To generate trustworthy pseudo labels based on the timestamp supervision, videos or video features are fed into the uncertainty estimation to obtain the uncertainty scores for each frame. Based on the uncertainty scores, we diffuse the timestamp annotation to the new pseudo labels. (c) Loop training. Due to the computation and memory cost, the loop training is introduced to optimize the spatial feature extractor and temporal feature extractor by generated pseudo labels in an iterative way.

many confident pseudo labels as possible by considering the temporal relationships among frames, while discarding pseudo labels with large uncertainty.

C. Uncertainty Estimation

In deep learning, neural networks may generate false predictions with a high probability, which is called epistemic uncertainty resulting from the model itself [48]. To estimate the uncertainty of the deep networks, Monte Carlo Dropout [49] is proposed to approximate the posterior distribution for uncertainty estimation. Ensembles [50] trains multiple networks independently on the entire dataset using random, and the predictions of multiple networks are averaged over an ensemble. Follow-up researchers majorly focus on improving the quality of the predicted uncertainty scores by inference-based methods [51]–[53] or auto-encoder based methods [54], [55]. Estimation of uncertainty has also been investigated for medical image classification and segmentation. Laves *et al.* [56] leverages Monte Carlo dropout at test time, and shows that error prediction is correlated with higher uncertainty in OCT classification. Leibig [57] uses Monte Carlo dropout to conduct uncertainty estimation and shows that uncertainty-informed decisions can improve diagnostic performance. Wang *et al* [58] utilize Monte Carlo dropout and test data augmentation to reduce overconfident error predictions in 3D brain tumor and 2D brain segmentation. Different from current methods [57], [58] that directly use Monte Carlo Dropout to estimate each sample individually, in our proposed UATD, the uncertainty of each frame is estimated based on the relation of itself, its nearby timestamp annotations and its adjacent frames in the temporal axis, which is motivated by the property of the surgical phase.

III. Method

A. Problem Definition

Let $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$ be a surgical video with T frames, where $\mathbf{x}_t$ is the t-th frame. Each surgical video is divided into several phases, and there is no overlapping among phases. Our goal is to learn a spatial feature extractor network $f(\cdot)$ and a temporal feature extractor $g(\cdot)$ that maps the frame $\mathbf{x}_t$ to a phase label, which is presented in Fig. 2 (a). In the full supervision, the frame-wise labels $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_T\}$ are available. However, in our timestamp supervision, given a video consisting of N phases, where $N << T$, only a single timestamp in each phase are annotated as $\mathbf{Y}_{ts} = \{\mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_N}\}$, where t_i is in the i-th phase, $\mathbf{y}_{t_i} \in \{1, 2, \dots, C\}$, and C is the total number of classes.

To perform surgical phase recognition with timestamp supervision, we propose an uncertainty-aware temporal diffusion (UATD) to generate trustworthy pseudo labels, denoted as $\bar{\mathbf{Y}}$, from the timestamp supervision $\mathbf{Y}_{ts}$ to optimize $f(\cdot)$ and $g(\cdot)$. The proposed UATD is shown in Fig. 2 (b); see Sec. III-B for details. Furthermore, we introduce the loop training, which optimizes $f(\cdot)$ and $g(\cdot)$ in an iterative way to reduce the memory cost and imbalance optimization; see Fig. 2 (c) and Section III-C for details.

B. Uncertainty-aware Temporal Diffusion

In timestamp supervision $\mathbf{Y}_{ts}$, *i.e.*, only a single label for each phase, the total number of positive frames is quite small and may be difficult to learn a robust model. Although we do not have full annotations, it is clear that the phases are long events consisting of consecutive frames. Motivated by this property of surgical videos, we propose the uncertainty-aware temporal diffusion (UATD) to diffuse the single labelled frame to its corresponding high confident (*i.e.*, low uncertainty) neighbour frames. In this way, we can introduce more frames acting as pseudo labels into the training process. Furthermore, the diffusion of frames is stopped by low-confident frames, which can avoid ambiguous annotations. The proposed UATD consists of two components: uncertainty estimation and temporal diffusion. In the following, we describe the two components respectively.

Uncertainty estimation. In UATD, we first need to estimate the uncertainty of each frame to find the high-confident ones for the single annotated frame. To this end, we introduce Monte Carlo Dropout [49], a simple yet efficient way, to evaluate the uncertainty of each frame. In Monte Carlo Dropout, given an input denoted as $\mathbf{z}$ and a network denoted as $o(\cdot)$, we feed z into $o(\cdot)$ with different dropout K times and obtain a set of class probabilities. This process can be formulated as:

$$\mathbf{P} = \{\mathbf{p}^k = o(\mathbf{z})\}_{k=1}^K, \tag{1}$$

where $\mathbf{p}^k \in \mathbb{R}^C$and $\mathbf{P} \in \mathbb{R}^{K \times C}$, C is the total number classes. Then, we average these K vectors of probability, which can be formulated as $\mu(\mathbf{P}) \in \mathbb{R}^C$, where $\mu(\cdot)$ is the mean function. After that, we obtain the class label for the input by:

$$c = \text{argmax}\ \mu(\mathbf{P}). \tag{2}$$

Finally, we use the standard deviation to measure the uncertainty of the obtained the class label, *i.e.*, c, which can be formulated as:

$$u = \sigma\ (\mathbf{P}_c), \tag{3}$$

where u is the uncertainty score for $o(\cdot)$ with the input of $\mathbf{z}$. The higher u indicates that the model $o(\cdot)$ predicts $\mathbf{z}$ to class c with lower confidence, and vice versa. In this paper, we need to evaluate the uncertainty of both the spatial and temporal feature extractors, which are defined in Section III-A.

To conduct the uncertainty estimation for the spatial feature extractor $f(\cdot)$, we add an extra classification head $h(\cdot)$ to $f(\cdot)$ as shown in Fig. 2 (c), denoted as $h(f(\cdot))$ to obtain the classification prediction for each frame $\mathbf{x}_t$. Let $o(\cdot) = h(f(\cdot))$ and $\mathbf{z} = \mathbf{x}_t$, and then we can obtain the uncertainty score μ_t for each frame $\mathbf{x}_t$ by using Eq. 1 to Eq. 3. Similarly, to conduct the uncertainty estimation for the spatial feature extractor $g(\cdot)$, we can easily set $o(\cdot) = g(\cdot)$ and $\mathbf{z} = \mathbf{f}_t$, obtaining the uncertainty score for each frame feature $\mathbf{f}_t$.

Temporal diffusion. After obtaining the uncertainty score μ_t, we use the temporal diffusion module to diffuse the current labels to more pseudo labels for training in the next iteration; see the iterative training details in Sec. III-C. To be specific, we treat the labeled frames as anchors and start diffusion from anchors to the adjacent frames on either side of them in the temporal dimension, which is illustrated in Algorithm 1. By the temporal diffusion, one frame would be introduced into the next iteration training only if the uncertainty score of it is lower than a threshold τ and the predicted class label equals to its nearby timestamp frame. In this way, the generated pseudo label would be high confidence, avoiding introducing noisy annotations. Note that in the obtained pseudo labels $\overline{\mathbf{Y}} = \{\overline{\mathbf{y}}_t\}_{t=1}^T$, $\mathbf{y}_t = 0$ means the t-th frame is not labelled.

Algorithm 1 Temporal Diffusion

Input: Uncertainty scores $\{\mu_t\}_{t=1}^T$, prediction $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_t\}_{t=1}^T$, timestamp annotation $\mathbf{Y}_{ts} = \{\mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_N}\}$, uncertainty threshold τ.
Output: Pseudo labels $\overline{\mathbf{Y}} = \{\overline{\mathbf{y}}_t\}_{t=1}^T$ for the next iteration.

```
▷ Diffusion for each phase
for $i = 1$ to $N$ do
    ▷ Diffusion for the left side
    for $t = t_{i-1}$ to $t_i$ do
        $\overline{\mathbf{y}}_t = \hat{\mathbf{y}}_t \cdot \mathbb{1}(u_t < \tau) \cdot \mathbb{1}(\hat{\mathbf{y}}_t = \mathbf{y}_{t_i})$
    end for
    ▷ Diffusion for the right side
    for $t = t_i$ to $t_{i+1}$ do
        $\overline{\mathbf{y}}_t = \hat{\mathbf{y}}_t \cdot \mathbb{1}(u_t < \tau) \cdot \mathbb{1}(\hat{\mathbf{y}}_t = \mathbf{y}_{t_i})$
    end for
end for
```

Algorithm 2 Loop training

Input: Video $\mathbf{X}$, timestamp annotation $\mathbf{Y}_{ts}$, initial spatial feature extractor $f(\cdot)$, initial spatial classifier $h(\cdot)$, initial temporal feature extractor $g(\cdot)$, uncertainty threshold τ, forward times K, times of temporal diffusion n, times of loop training m.
Output: Well optimized spatial feature extractor $f(\cdot)$ and temporal feature extractor $g(\cdot)$.

```
$\overline{\mathbf{Y}} \leftarrow \mathbf{Y}_{ts}$                    ▷ Set the initial pseudo labels
for $i = 1$ to $n$ do
    ▷ Optimizing the spatial feature extractor
    $f(\cdot), h(\cdot) \leftarrow$ OptimS$(f(\cdot), h(\cdot), \mathbf{X}, \overline{\mathbf{Y}}, \mathcal{L}_{ce})$
    ▷ Use UATD to generate the new pseudo labels
    $\overline{\mathbf{Y}} \leftarrow$ UATD$(h(f(\cdot)), \mathbf{X}, \mathbf{Y}_{ts}, \tau, K)$.
end for
for $j = 1$ to $m$ do
    $\mathbf{F} \leftarrow f(\mathbf{X})$
    for $i = 1$ to $n$ do
        ▷ Optimizing the spatial feature extractor
        $g(\cdot) \leftarrow$ OptimT$(g(\cdot), \mathbf{F}, \overline{\mathbf{Y}}, \mathcal{L}_{ce}, \mathcal{L}_{smooth})$
        ▷ Use UATD to generate the new pseudo labels
        $\overline{\mathbf{Y}} \leftarrow$ UATD($g(\cdot)$, $\mathbf{F}$, $\mathbf{Y}_{ts}$, $\tau$, $K$)
    end for
    ▷ Optimizing the spatial feature extractor
    $f(\cdot), h(\cdot) \leftarrow$ OptimS$(f(\cdot), h(\cdot), \mathbf{X}, \overline{\mathbf{Y}}, \mathcal{L}_{ce})$
end for
```

For clarity, we formulate the overall process of UATD as $\overline{\mathbf{Y}} \leftarrow$ UATD$(o(\cdot), \mathbf{Z}, \hat{\mathbf{Y}}, \tau, K)$, where $\mathbf{Z}$ is the input (*e.g.*, $\mathbf{X}$ or $\mathbf{F}$), $o(\cdot)$ is the network (*e.g.*, $h(f(\cdot))$ or $g(\cdot)$).

C. Loop Training

The duration of the surgical videos generally lasts tens of minutes or even hours, making it hard to train the model in the end-to-end manner. In previous full supervised methods [3], [4], [15], a few consequent frames are sampled from the long videos for training the spatial and temporal networks in the end-to-end manner. However, in timestamp annotation, most of the sampled frames have no labels, resulting in the imbalance of positive and negative samples. This imbalance training would degrade the performance; see details in Table III. To address this problem, we decouple the optimization of spatial and temporal feature extractors via loop training, as shown in Fig. 2 (c). In the loop training, we only sample labelled frames (annotated timestamps or generated pseudo labels) to optimize the spatial feature extractor or temporal feature extractor, which can not be achieved in previous joint training. Formally, there are four main steps in our loop training:

(a) Optimizing the spatial feature extractor: $f(\cdot), h(\cdot) \leftarrow$ OptimS$(f(\cdot), h(\cdot), \mathbf{X}, \overline{\mathbf{Y}}, \mathcal{L}_{ce})$. To be specific, the input video $\mathbf{X}$ is fed into the spatial feature extractor f to obtain the video feature $\mathbf{F} = f(\mathbf{X})$. Then a classifier $h(\cdot)$ is used to obtain the prediction $\hat{\mathbf{Y}} = h(\mathbf{F})$, where $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_t\}_{t=1}^{T}$. Given the target labels (timestamp annotation or pseudo labels) $\overline{\mathbf{Y}} = \{\overline{\mathbf{y}}_t\}_{t_1}^{T}$, the objective for the spatial feature extractor can be formulated as:

$$\mathcal{L}_{ce} = -\frac{1}{T}\sum_{t=1,\overline{\mathbf{y}}_t\neq 0}^{T} \overline{\mathbf{y}}_t \log(\hat{\mathbf{y}}_t), \tag{4}$$

where $\overline{\mathbf{y}}_t \neq 0$ indicates the t-th frame is not labelled.

(b) Extracting the spatial features: $\mathbf{F} = f(\mathbf{X})$; see details in step **(a)**.

(c) Optimizing the temporal feature extractor: $g(\cdot) \leftarrow$ OptimT$(g(\cdot), \mathbf{F}, \overline{\mathbf{Y}}, \mathcal{L}_{ce}, \mathcal{L}_{smooth})$. Specifically, the video feature $\mathbf{F}$ is fed into $g(\cdot)$ to capture the temporal relation of frames and obtain the corresponding predictions $\hat{\mathbf{Y}}$. We use the CrossEntropy loss to train the $g(\cdot)$, similar to $f(\cdot)$. Compared with the spatial feature extractor, to encourage a smooth transition between frames, we use the truncated mean squared error as a Smoothing Loss following [14], [59]:

$$\mathcal{L}_{smooth} = \frac{1}{TC}\sum_{t,c}\tilde{\Delta}_{t,c}^{2}, \tag{5}$$

$$\tilde{\Delta}_{t,c}^{2} = \begin{cases} \Delta_{t,c}^{2}, & \Delta_{t,c} < \gamma \\ \gamma, & otherwise \end{cases}, \tag{6}$$

$$\Delta_{t,c} = |log(\boldsymbol{\hat{y}}_{t,c}) - log(\boldsymbol{\hat{y}}_{t-1,c})|, \tag{7}$$

where T is the video length and C is the number of action classes. This loss function explicitly penalizes the difference between every two adjacent frames and we suggest readers refer to [59] for more details. The final loss function is the weighted sum of these two losses:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{smooth}, \tag{8}$$

where λ is a hyper-parameter to balance the contribution of each loss and is set to 0.015 for all of our experiments.

(d) Generating the pseudo labels based on UATD: $\overline{\mathbf{Y}} \leftarrow$ UATD$(o(\cdot), \mathbf{Z}, \mathbf{Y}_{ts}, \tau, K)$; see details in Section III-B.

After the definition of the four steps, we illustrate the loop training in Algorithm 2.

IV. Experiments

A. Datasets and Metrics

M2CAI16. The M2CAI16 dataset [61] consists of 41 laparoscopic videos that are acquired at 25fps of cholecystectomy procedures, and each frame has a resolution of 1920×1080. Following [21], 27 videos are used for training while the rest 14 are used for testing. These videos are segmented into 8 phases by experienced surgeons.

Cholec80. The cholec80 dataset [5] contains 80 videos of cholecystectomy surgeries performed by 13 surgeons. All the videos are recorded at 25 fps, and the frames in them have a resolution of 1920×1080 or 854×480. The dataset is divided into two subsets of equal size, with the first 40 videos as a training set and the other 40 as a testing set.

Evaluation metrics. Following previous works [3]–[5], [20], we utilize four metrics, *i.e.*, accuracy (AC), precision (PR), recall (RE), and Jaccard (JA), to evaluate the phase prediction accuracy. Among them, accuracy and Jaccard index are used to evaluate the submission of M2CAI Workflow Challenge, while precision and recall are also commonly used metrics for video-based phase recognition.

B. Annotation Analysis

To obtain the timestamp annotations, we invite two surgeons to label a single timestamp for each phase on two datasets. Specifically, they are asked to label one timestamp for each phase while watching the video, as shown in Fig. I (a). To compare the annotation cost of different types of annotations, we also ask them to find the precise start and end time for each phase, *i.e.*, full annotation. In Fig. 3 (a), we report the average time they spend on each video when using timestamp and full annotations. “Surgeon1” and “Surgeon2” indicates the first surgeon and the second surgeon respectively. To obtain annotation times for full or timestamp annotations, we first let the surgeon prepare a timer. When conducting full or timestamp annotations, the surgeon first turned on the timer, then immediately watched the video and annotated it. After completing the annotation of a video, the surgeon stopped the timer immediately, and record the time it takes to annotate the video. When all videos are annotated and their annotation time are recorded, we calculate the average annotation time for all videos. It is clear that our introduced timestamp annotation can largely reduce the annotation time compared with the full annotation, *e.g.*, Surgeon2 can reduce 78% time in Cholec80 dataset. On average, our proposed timestamp annotation only requires 26% annotation time compared with the full annotation.

Furthermore, we also show the distribution of the relative position of timestamp annotation to the corresponding phase on two datasets. As shown in Fig. 3 (b), the labeled timestamps would appear in an arbitrary position of the phase. Surgeons prefer to label timestamps near the middle of phases, which reveals that surgeons can identify a phase without watching the whole phase. That is to say, the surgeons can skip the left part of the phase after the timestamp annotation. Of course, there is no need for the surgeons to repeat watching videos to find the precise temporal window for each phase. Hence, the annotation time can vastly be reduced compared with the full annotation. In the implementation, one second of video is converted to 25 frames. To save memory and computation cost, we sample one frame every second. Hence, during annotation, the surgeon labels the second, and during implementation,

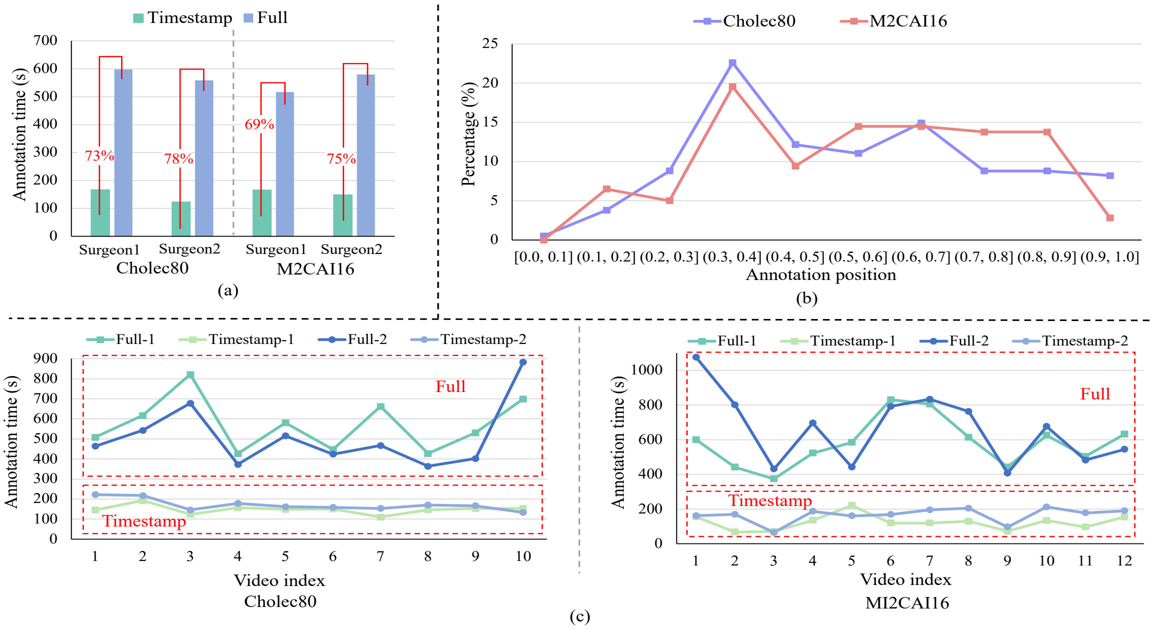


Fig. 3. (a) Comparison of the annotation time of timestamp and full annotations. We show the annotation times (seconds/video) of two different surgeons for Cholec80 and M2CAI16 datasets, respectively. (b) Statistics of positions of manually annotated timestamps on two datasets. The horizontal axis indicates the relative temporal portion of the whole phase. For example, (0.1, 0.2] indicates the annotated timestamp is inside the temporal period from 0.1 to 0.2 of the phase. The vertical axis represents the percentage of annotated timestamps. It shows that the timestamps would appear in the arbitrary position of the phase, and surgeons prefer to label timestamps near the middle of phases. (c) Statistics of annotation time of manually annotated timestamps on two datasets. The horizontal axis indicates the video index, and the vertical axis represents annotation time. It shows that our timestamp annotation consistently takes less time than full annotation for labeling each video.

TABLE I

COMPARISON WITH THE STATE-OF-THE-ART ON CHOLEC80 AND M2CAI16 DATASETS. "*" INDICATES THE OFFLINE PREDICTION.

Method	Cholec80				M2CAI16			
	AC (%)	PR (%)	RE (%)	JA (%)	AC (%)	PR (%)	RE (%)	JA (%)
Fully Supervised Methods - **100**% annotation time								
PhaseNet [60]	78.8 ± 4.7	71.3 ± 15.6	76.6 ± 16.6	-	79.5 ± 12.1	-	-	64.1 ± 10.3
EndoNet [5]	81.7 ± 4.2	-	79.6 ± 7.9	-	-	-	-	-
SV-RCNet [3]	85.3 ± 7.3	80.7 ± 7.0	83.5 ± 7.5	-	81.7 ± 8.1	81.0 ± 8.3	81.6 ± 7.2	65.4 ± 8.9
OHFM [20]	87.3 ± 5.7	-	-	67.0 ± 13.3	85.2 ± 7.5	-	-	68.8 ± 10.5
Casual TCN [14]	87.9 ± 8.2	86.4 ± 7.7	84.8 ± 12.9	72.4 ± 9.4	81.9 ± 11.3	84.8 ± 5.2	82.2 ± 9.0	68.1 ± 8.5
TeCNO [25]	88.6 ± 7.8	86.5 ± 7.0	88.8 ± 17.4	75.1 ± 6.9	-	-	-	-
TMRNet [4]	90.1 ± 7.6	90.3 ± 3.3	89.5 ± 5.0	79.1 ± 5.7	87.0 ± 8.6	87.8 ± 6.9	88.4 ± 5.3	75.1 ± 6.9
Trans-SVNet [13]	90.3 ± 7.1	90.7 ± 5.0	88.8 ± 7.4	79.3 ± 6.6	87.2 ± 9.3	88.0 ± 6.7	87.5 ± 5.5	74.7 ± 7.7
TCN* [59]	91.1 ± 6.7	90.8 ± 4.5	87.6 ± 11.7	79.1 ± 8.5	82.9 ± 10.8	85.8 ± 5.4	82.7 ± 9.0	69.7 ± 8.7
Not end-to-end [21]	91.5 ± 7.1	-	87.2 ± 8.2	77.2 ± 11.2	88.2 ± 8.5	-	91.4 ± 11.2	75.1 ± 10.6
Semi Supervised Methods - **50**% annotation time								
LRTD [32]	82.5 ± 8.4	79.7 ± 9.0	80.9 ± 8.1	64.2 ± 10.2	72.1 ± 13.7	74.1 ± 14.9	74.0 ± 10.4	54.4 ± 12.9
SurgSSL [33]	87.0 ± 7.4	84.2 ± 8.9	85.2 ± 11.1	70.5 ± 12.6	79.6 ± 9.4	80.2 ± 11.3	79.6 ± 11.5	62.0 ± 11.1
Timestamp Supervised Methods - **26**% annotation time								
Casual TCN+Ours	88.6 ± 6.7	86.1 ± 6.7	88.0 ± 10.1	73.7 ± 10.2	86.0 ± 7.8	85.0 ± 6.2	87.1 ± 7.7	71.4 ± 10.4
TCN*+Ours	91.9 ± 5.6	89.5 ± 4.4	90.5 ± 5.9	79.9 ± 8.5	87.6 ± 8.7	88.2 ± 7.4	87.9 ± 9.6	75.7 ± 9.5

we set the frame belonging to the timestamp second as the annotation. Finally, we show the statistics of annotation time of manually annotated timestamps on two datasets in Fig. 3 (c). The results show that the annotation times of timestamps are much less than those of full for all videos.

C. Implementation Details

Our code is based on PyTorch using an NVIDIA GeForce RTX 3090 GPU. We downsample the video to 1fps for training in all experiments following previous works [3]–[5]. All the frames are resized to a resolution of 250 × 250, and data augmentations including 224 × 224 cropping, random mirroring, and color jittering are applied during the training stage. We get a pre-trained inception-v3 [62] on ImageNet [63]. The batch size is set to 8, and an Adam optimizer with an initial learning rate of $1e-4$ and weight decay of $1e-5$ is used. We further use a step learning rate scheduler where the step size is two epochs and the decay rate is 0.5 for fine-tuning

TABLE II

COMPARISON WITH DIFFERENT TIMESTAMP SUPERVISION METHODS.

Method	AC (%)	PR (%)	RE (%)	JA (%)
		Cholec80		
Naive	66.9 ± 5.6	62.3 ± 6.5	74.8 ± 6.5	48.2 ± 4.4
Uniform	58.7 ± 7.8	55.5 ± 6.1	65.9 ± 5.4	39.0 ± 6.5
Li *et al.* [14]	79.4 ± 5.5	78.7 ± 6.5	85.4 ± 5.5	64.0 ± 5.6
Ours	88.6 ± 6.7	86.1 ± 6.7	88.0 ± 10.1	73.7 ± 10.2
		M2CAI16		
Naive	67.5 ± 7.2	58.7 ± 6.5	61.7 ± 6.5	44.8 ± 6.7
Uniform	56.5 ± 8.7	56.7 ± 7.7	57.0 ± 7.9	38.2 ± 5.6
Li *et al.* [14]	72.7 ± 8.8	76.5 ± 7.1	80.5 ± 6.9	59.9 ± 10.1
Ours	86.0 ± 7.8	85.0 ± 6.2	87.1 ± 7.7	71.4 ± 10.4

by 5 epochs. To train TCN, we use Adam optimizer with an initial learning rate of $1e-3$ and cosine annealing for learning rate decay. For all experiments, we set a dropout rate of 0.5 and an uncertainty threshold $\tau = 0.1$; the detailed analysis is shown in Table IV. The uncertainty is estimated by 5 forward times Monte Carlo Dropout. [49]. The numbers of rounds of uncertainty-aware temporal diffusion and loop training are set to $m = 4$ and $n = 2$, respectively. Furthermore, the timestamp annotations are simulated by randomly selecting one frame from each action phase in the training videos.

D. Comparison with the State-of-the-Arts

We compare our *less is more* method with the state-of-the-arts on the Cholec80 and M2CAI16 datasets, and report their results in Table I. The numbers in Table I are the mean and standard deviation of performance of all phases. For example, in Cholec80, there are 7 phases. To obtain the accuracy (AC) for each method, we first obtain AC for each phase, the mean of AC of 7 phases is computed to obtain the first number in Table I. After that, we compute the standard deviation of AC numbers of 7 phases to obtain the number after "+/-". The computation of PR, RE and JA is like AC. It is clear that our method outperforms previous data-efficient methods, *i.e.*, semi-supervised ones, on both data efficiency and phase recognition performance. For example, our timestamp supervision only requires 26% annotation time of the full supervision [47], while semi-supervision needs 50% annotation time [32]. Moreover, our method with the casual TCN [14] achieves 88.6% of accuracy on Cholec80 dataset, achieving the competitive performance compared to semi-supervised methods. We can also find that our method can even achieve the competitive performance compared with the fully supervised methods, with only 26% annotation time of them. Notably, the improvements of our method are more significant in M2CAI16 than in Cholec80. This is because M2CAI16 contains more ambiguous frames [12], which degrades the performance. We also illustrate the comparison of per-phase performance with fully supervised methods in Fig. 5. The results show that our methods achieve substantial improvement for all phases. The details of why our methods can outperform corresponding backbones in the fully supervised setup will be discussed in Sec. IV-F.

TABLE III

ABLATION STUDY OF KEY COMPONENTS ON CHOLEC80 DATASET. 'UATD (S)' AND 'UATD (T)' INDICATE USING UATD IN THE SPATIAL FEATURE AND TEMPORAL FEATURE EXTRACTORS. 'LP' INDICATES THE LOOP TRAINING WHICH IS DEFINED IN SEC. III-C.

UATD (S)	UATD (T)	LP	AC (%)	PR (%)	RE (%)	JA (%)
✗	✗	✗	66.9 ± 9.7	62.3 ± 6.7	74.8 ± 7.8	48.2 ± 7.6
✓	✗	✗	82.3 ± 7.6	78.1 ± 8.8	86.9 ± 6.5	66.0 ± 7.4
✗	✓	✗	77.6 ± 5.3	77.3 ± 6.7	81.0 ± 7.3	61.3 ± 5.2
✗	✗	✓	68.5 ± 4.8	63.7 ± 6.2	75.2 ± 3.7	50.2 ± 6.1
✓	✓	✗	85.6 ± 7.4	83.5 ± 6.5	86.6 ± 6.1	70.9 ± 8.2
✓	✓	✓	88.6 ± 6.7	86.1 ± 6.7	88.0 ± 10.1	73.7 ± 10.2

E. Comparison with Different Timestamp Supervision Methods

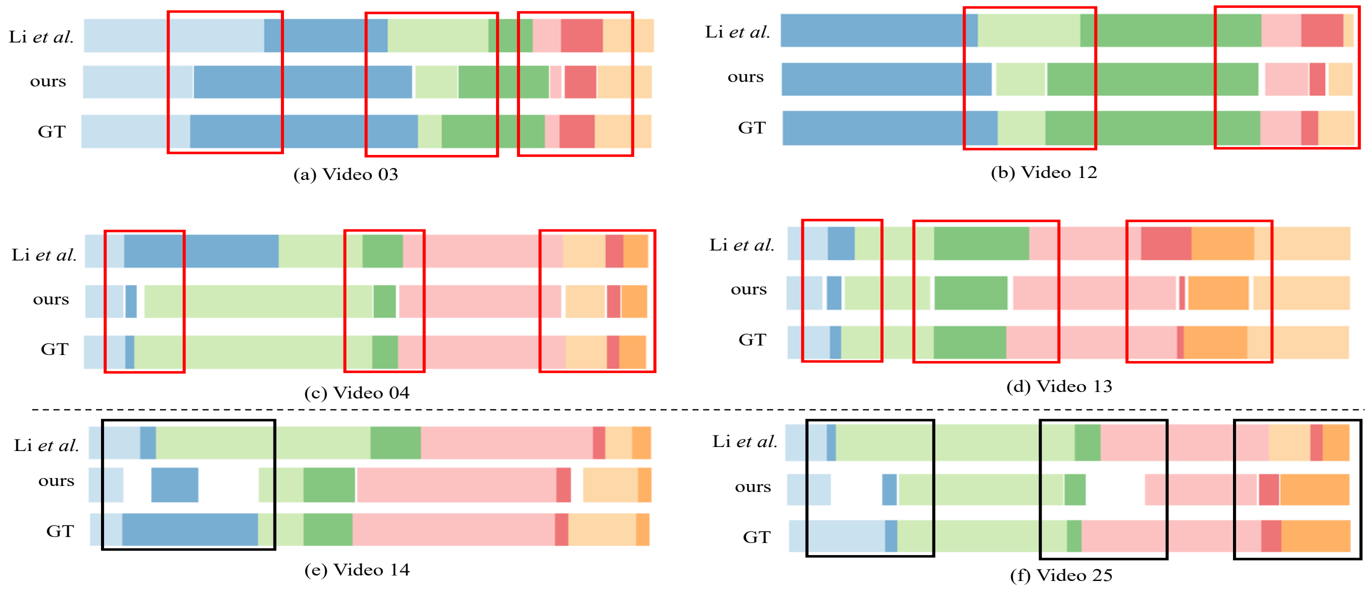
To evaluate the efficiency of our proposed uncertainty-aware temporal diffusion (UATD) for surgical video timestamp supervision, we compare our methods with three baseline models, *i.e.*, Naive, Uniform and Li *et al.* [14], and report the results in Table II. Specifically, in Naive, we only use the annotated timestamp labels to supervise the model training, without generating any pseudo labels. In Uniform, the pseudo labels are generated by a uniform way, *i.e.*, the action labels change at the center frame between two timestamp annotations. For example, assuming two timestamps $\mathbf{y}_{t_1}$ and $\mathbf{y}_{t_2}$ with $t_1 < t_2$, then the pseudo labels can be generated as:

$$\hat{\mathbf{y}}_t = \begin{cases} \mathbf{y}_{t_1}, & t \in (t_1, t_1 + (t_2 - t_1)/2] \\ \mathbf{y}_{t_2}, & t \in (t_1 + (t_2 - t_1)/2, t_2) \end{cases}. \tag{9}$$

It is clear that our method outperforms the other two methods by a clear margin. Furthermore, we also compare Li *et al.* [14], which is the SOTA in action segmentation under this setting. Specifically, Li *et al.* [14] aims to identify changes of actions to divide the videos into segments. For each action change, the frames before the change should be assigned to the class label of the previous timestamp and after the change to the next timestamp, hence generating pseudo labels for all frames. This method performed well in cases where there are clear boundaries between different phases, like natural videos, but generated worse results in surgical videos. This is mainly attributed to the fact that compared to natural videos, surgical videos have many ambiguous boundaries, leading to degraded performance. Therefore, we propose a novel method to generate pseudo labels for confident frames, effectively avoiding erroneous pseudo labels near boundaries. As shown in Fig. 4, the pseudo labels generated from [14] contain much noise near the boundaries of each action, which may affect the retraining of the network. Unlike [14], our method generates more accurate and high-quality pseudo labels, thus achieving better performance (see Table II). Furthermore, our method can avoid generating low-confidence and noisy pseudo labels, thus improving the robustness of the models. From Table. II, we can also see that our method obtains $7\% - 13\%$ improvements over all metrics.

F. Ablation Study

Effect of UATD and LP. There are two key components, *i.e.*, uncertainty-aware temporal diffusion (UATD) and loop



(a) Video 03

(b) Video 12

(c) Video 04

(d) Video 13

(e) Video 14

(f) Video 25

Fig. 4. Comparison of the visualization of pseudo labels generated by ours and Li *et al.* [14]. "GT" indicates the ground-truth. We sample four videos, *i.e.*, from Cholec80 ((a)-(b)) and M2CAI16 ((c)-(d)). It is clear that our method can generate more accurate pseudo labels compared with Li *et al.* [14]; see red boxes. We illustrate the worst pseudo labels generated by our method shown in (e)-(f). Compared with Li *et al.* [14] which brings erroneous pseudo labels into training, our method would avoid generating low confident labels, *i.e.*, the uncertainty frames inside the phases would make the model stop temporal diffusion (see black boxes).

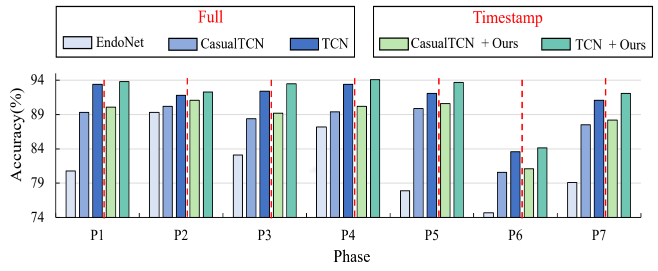


Fig. 5. Comparison of per-phase performance with fully supervised approaches on Cholec80 dataset. 'P1' indicates the first phase and so on. For each phase, the three methods on the left of the red line are fully supervised methods, and the two on the right are our proposed timestamp methods.

TABLE IV

QUANTITATIVE RESULTS OF DIFFERENT UNCERTAINTY THRESHOLDS.

	AC (%)	PR (%)	RE (%)	JA (%)	Labelling Rate (%)	Labelling Accuracy (%)
$\tau = \infty$	84.65	86.34	84.65	70.43	**93.49**	92.04
$\tau = 0.2$	85.32	**86.71**	85.06	71.14	92.72	94.13
$\tau = 0.1$	**85.95**	84.96	**87.05**	**71.43**	87.51	96.99
$\tau = 0.05$	85.49	86.19	86.42	71.20	60.76	**99.04**

training (LP), in our method. We ablate the effect of them in Table. III. It is clear that the proposed UATD can improve the timestamp supervision with a clear margin, *e.g.*, combined with UATD, the model achieves 85.6% accuracy, outperforming 18.7% over the baseline model. Futhermore, we could find that LP (the fourth row in Table III) contributes to around 3% improvements in AC, compared with the baseline (the first row in Table III). The combination of UATD and LP can boost the performance to 88.6%.

Impact of the uncertainty threshold τ. The quality of pseudo labels is dependent on pseudo labeling rate and pseudo labels accuracy, which is controlled by the uncertainty threshold τ in Algorithm 1. In order to evaluate the effect of τ, we compare the performance of the models with different τ and report the results in Table IV. "Labelling Rate" indicates the ratio of the frames annotated by our method to all frames. To evaluate the accuracy of our generated annotations, *i.e.*, pseudo labels, we compare the generated pseudo labels with the ground-truth. Specifically, for a frame, if the annotated label generated by

TABLE V

COMPARISON OF LABELLING RATE AND LABELLING ACCURACY OF PSEUDO LABELS GENERATED BY UATD IN DIFFERENT ITERATIONS. "TS" INDICATES THE INITIAL TIMESTAMP ANNOTATIONS.

Iteration	TS	1-st	2-nd	3-rd
Labelling Rate (%)	0.33	67.70	76.82	84.45
Labelling Accuracy (%)	100.00	98.69	97.95	97.42

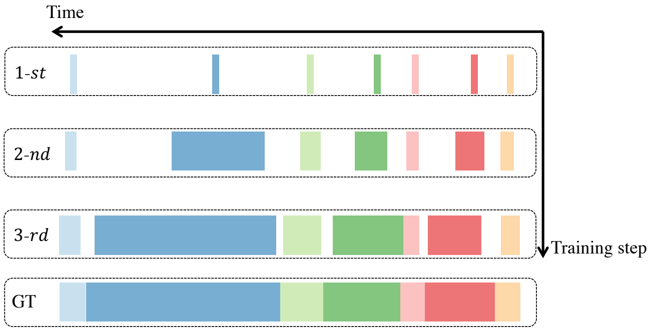
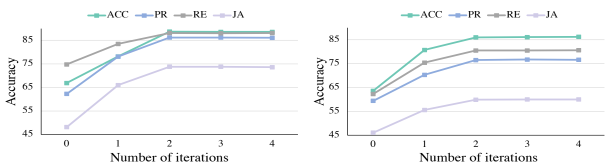
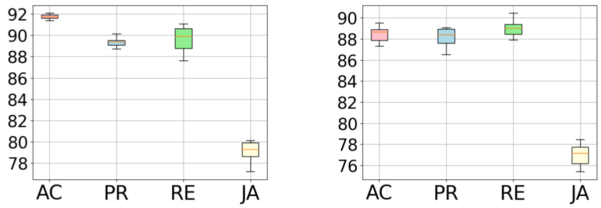


Fig. 6. Visualization of the different iterations of the pseudo labels generated by our method. "GT" indicates the ground truth. "**1**-st", "2-nd" and "**3**-rd" indicate generated pseudo labels in the first, second and third iterations respectively.



(a) (b)

Fig. 7. Analysis of the number of iterations for loop training on (a) Cholec80 and (b) M2CAI16. We show the results of ACC, PR, RE and JA of models with different numbers of iterations.



(a) Cholec80 (b) M2CAI16

Fig. 8. Box plots of performance of random timestamp annotations on Cholec80 and M2CAI16 datasets.

our method is equal to the ground-truth, the frame is regarded as the correct annotated frame, and vice versa. We can find that the higher uncertainty threshold would lead to a higher pseudo labeling rate and the lower accuracy of pseudo labels, and vice versa. For example, with infinity threshold, *i.e.*, first row in Table IV, pseudo labeling rate can reach 93.49% while accuracy of pseudo labels is only 92.04%. Such a higher labeling rate would introduce more noisy labels, which degrades the labeling accuracy. Furthermore, with different τ, *i.e.*, 0.2, 0.1 and 0.05, the performance of our method is very stable. For example, the variance for accuracy values with different thresholds is only 0.11% In our paper, we set τ to 0.1 for the best trade-off.

Analysis of pseudo labels in different iterations. Given only a single manual labeled annotations, we show that our model can generate more and more reliable pseudo-labels step by step in Table V. "Labelling Rate" and "Labelling Accuracy" are the same meaning as Table IV. It shows that our method can generate more and more pseudo labels as the number of iterations increases. This is because each iteration of temporal diffusion gives the temporal model extra information, and the model can generate more pseudo labels next time. Also, the accuracy of generated pseudo labels is very trustworthy. Since the frames show very similar appearances to their adjacent frames, the network can easily generate correct predictions for the neighbor frames of the annotated frame. We also show the visualization of the different iterations of the pseudo labels generated by our method in Fig. 6.

Effect the number of iterations for loop training. We conduct the analysis of the number of iterations for loop training in Fig. 7. The results show that more iterations for loop training can improve the performance, since more trustworthy labels are introduced to training. We also find that there is no significant performance improvement after more than two iterations. Hence, in this paper, we set the number of iterations for loop training as two.

TABLE VI

QUANTITATIVE RESULTS OF START, END, MIDDLE AND RANDOM TIMESTAMP POSITIONS ON CHOLEC80 DATASET.

Timestamp Position	AC (%)	PR (%)	RE (%)	JA (%)
Start	90.64	87.92	88.37	76.75
End	90.17	88.35	82.24	70.75
Middle	**92.59**	**90.13**	**89.60**	**80.04**
Random	91.86	89.51	90.52	79.90

TABLE VII

COMPARISON OF ANNOTATION TIME BETWEEN A SINGLE TIMESTAMP AND TWO TIMESTAMPS.

Video Index	01	05
Single Timestamp	222s	155s
Two Timestamps	331s	279s

Robust to different timestamp annotations. In our experiments, the timestamp annotations are generated by randomly selecting one frame to be annotated for each phase. In order to evaluate whether our method is robust to the different timestamps, we random 10 different timestamps by different random seeds and analyse their impacts on the performance, which is shown in Fig. 8. Specifically, we report the box plots of 10 random timestamp annotations on Cholec80 and M2CAI16 datasets. The short and flat boxes indicate that our proposed method is robust to different timestamp annotations, *e.g.*, the difference between the maximum and minimum is 2.3%. What's more, our method can outperform most of the methods in Table. I with even the worst timestamp annotations.

Effect of timestamps in different phase positions. In order to explore the effect of timestamps in different phase positions, we enforce the random timestamp annotations inside the start, end or middle region of each phase. More specifically, we regard the first 10% frames, the middle 10% frames and the last 10% frames of each phase as the start, middle and end regions. As shown in Table. VI, annotating at the start and end frames of each phase would degrade the performance. This is because that frames near boundaries are generally ambiguous, which can be hard to act as an anchor of temporal diffusion. In the contrast, the middle frames are more discriminative to represent current phases and thus can generate more correct pseudo labels. Actually, the surgeons ,*i.e.*, the annotators, tend to label the discriminative frames because they can easily recognize them when seeing through the whole video [47], which ensures timestamp annotations efficiently and effectively.

Comparison between a single and two timestamps. During the timestamp annotation, once a phase is identified and the current timestamp is recorded, the surgeon could choose to record another timestamp for the phase. Here, we compare the annotation cost between a single and two timestamps in Table VII. Two videos, *i.e.*,"01" and "05", are sampled from Cholec80 and M2CAI16 respectively. The result shows that two timestamp annotations would cost more time than a single timestamp annotation, *e.g.*, the surgeon would spend 331s for "01" while annotating a single timestamp only requires 222s.

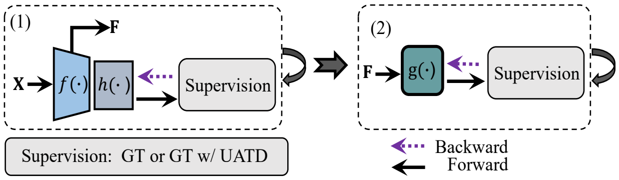


Fig. 9. Loop training for GT or GT masked with UATD, which consists of two steps: the optimization of (1) the spatial feature extractor and (2) the temporal feature extractor. Different from the dynamic pseudo labels generated by UATD, GT or GT w/ UATD is fixed during training. Hence, there is no need for iteratively optimizing the spatial feature extractor and the temporal feature extractor. Here, we first train the spatial feature extractor n times individually and obtain the fixed spatial features from the well-trained spatial feature extractor. Then, the temporal feature extractor is optimized m times separately. We set $n = 2$ and $m = 4$ as the loop training for pseudo labels generated by UATD (see Section IV-C).

TABLE VIII
COMPARISON OF PERFORMANCE OF MODELS TRAINING WITH A SINGLE TIMESTAMP AND TWO TIMESTAMPS.

Method	Annotation	AC (%)	PR (%)	RE (%)	JA (%)
Cholec80					
Casual TCN	Single	88.56	86.05	88.00	73.72
	Two	88.79	89.61	88.12	73.80
TCN	Single	91.86	89.51	90.52	79.90
	Two	91.91	89.66	90.81	79.93
M2CAI16					
Casual TCN	Single	86.03	85.02	87.08	71.43
	Two	86.07	85.11	87.14	71.50
TCN	Single	87.62	88.25	87.91	75.72
	Two	87.70	88.30	87.98	**75.81**

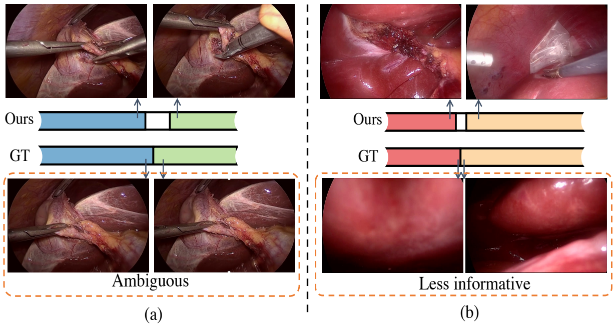


Fig. 10. Comparison of pseudo labels generated by ours and ground-truth. It is clear that our method avoids annotating the frames near boundaries, where frames are generally (a) ambiguous or (b) less informative. In our paper, we regard ambiguous frames as the frames that show similar appearance in different phases following [12]. Less informative frames indicate the frames that provide little information to identify different phases, such as phases containing no actions or instruments.

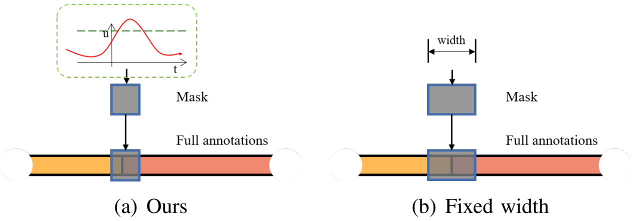


Fig. 11. (a) Masking boundaries by using UATD to detect ambiguous frames. (b) Masking boundaries by fixed width.

We also conduct experiments to compare the performance of the models training with a single timestamp and two timestamps, as shown in Table VIII. The results show that two timestamp annotations cannot achieve clear improvement but bring additional annotation costs. Hence, annotating a single timestamp is much more efficient than two timestamps, and we use the best efficient way to solve surgical phase recognition in this paper.

Comparison of generated pseudo label and ground-truth. In our experiments, we find that our method only generates pseudo labels for discriminative frames while ignoring the ambiguous ones near boundaries. As shown in Fig. 10, our generated pseudo labels discard ambiguous or less informative frames compared to the ground-truth. More importantly, the model trained with our generated pseudo labels outperforms the model trained with the ground-truth; see details in Table. I. This indicates that the ambiguous boundary of two adjacent actions would degree the performance.

G. Incorporate UATD into Current Methods

As analyzed in Fig. 10, we find that our method can only generate labels for discriminative frames, instead of ambiguous frames. It comes up that if masking ambiguous frames from the ground-truth by our UATD can improve the performance. To this end, as shown in Fig. 11 (a), we mask some ground-truth labels near boundaries, based on the pseudo labels generated by our methods. To be specific, we use UATD to generate pseudo labels, and record the indexes of unlabelled frames that are with high uncertainty. Then, we remove those frames with the recorded indexes from the ground-truth, and obtained a clean ground-truth to supervise the model. Note that we use the pseudo labels generated in the final iteration (see details in Section IV-C) to mask the ground-truth. We regard the obtained clean ground-truth as 'GT W/ UATD' in the following. We compare the performance of the models training with **(a)** ground-truth (GT), **(b)** pseudo labels generated by UATD and **(c)** GT W/ UATD, and report the results in Table IX. The details for training pseudo labels generated by UATD are illustrated in Fig. 2 and Section III-C. The training of GT W/ UATD is the same as the full supervised training with GT, which is shown in Fig. 9. To show the generalization of our proposed method, we also conduct experiments on JIGSAW [64], which is a simulated dataset with a clear domain gap. From Table IX, it is clear that the model training with GT masked by UATD can achieve the best results, and even outperforms the current SOTA; see Table I for comparison. We further conduct experiments on the models training with GT masked by the fixed width, as shown in Fig. 11 (b). As illustrated in Table. X, masking some frames near the boundary during training outperforms the model without masking over around $1\% - 3\%$ in all metrics. However, it will introduce a new hyper-parameter, *i.e.*, the width of the mask, which is critical to the performance. Hence, in order to achieve good performance, we need to conduct many experiments to find the best choice, which is very time-consuming. *On the contrary, our method can be used as an approach to clean the noisy labels in the ground-truth automatically, without the need for hand-designed width.*

TABLE IX

EFFECTIVENESS OF INCORPORATING UATD INTO CURRENT METHODS. 'TIMESTAMP' IS USING OUR GENERATED PSEUDO LABELS BY UATD FROM TIMESTAMP ANNOTATIONS AND 'GT W/ UATD' INDICATES THE GROUND-TRUTH LABELS MASKED BY UATD; SEE SEC. IV-G FOR DETAILS.

Method	Annotation	AC (%)	PR (%)	RE (%)	JA (%)
Cholec80					
Casual TCN	GT	87.94	86.40	84.81	72.40
	Timestamp	88.56	86.05	88.00	73.72
	GT w/ UATD	**91.18**	**89.88**	**90.93**	**79.76**
TCN	GT	91.14	90.84	87.64	79.14
	Timestamp	91.86	89.51	90.52	79.90
	GT w/ UATD	**92.75**	**91.23**	**93.10**	**83.89**
M2CAI16					
Casual TCN	GT	81.91	84.82	82.24	68.06
	Timestamp	86.03	85.02	87.08	71.43
	GT w/ UATD	**87.01**	**88.23**	**88.81**	**76.26**
TCN	GT	82.94	85.82	82.69	69.71
	Timestamp	87.62	88.25	87.91	75.72
	GT w/ UATD	**88.32**	**89.03**	**89.23**	**78.81**
JIGSAW					
Casual TCN	GT	80.12	82.02	81.16	69.11
	Timestamp	81.73	84.91	84.82	71.55
	GT w/ UATD	**83.27**	**85.51**	**85.27**	**72.81**
TCN	GT	81.43	84.29	83.71	70.18
	Timestamp	83.18	85.19	85.72	72.12
	GT w/ UATD	**84.28**	**86.13**	**86.16**	**73.15**

TABLE X

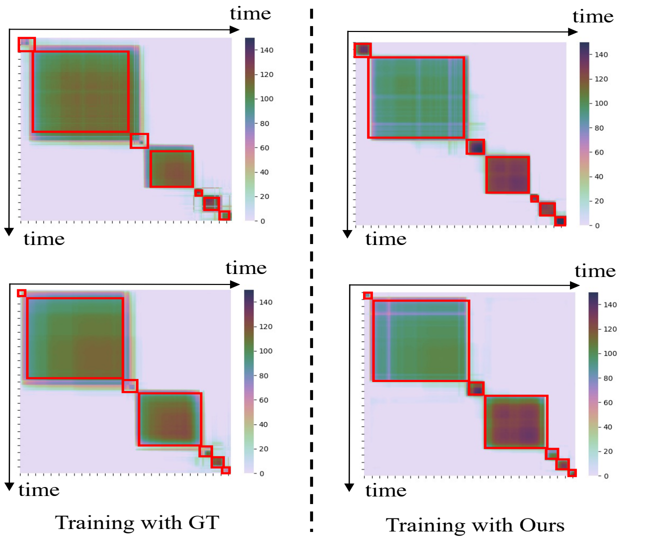
EFFECTIVENESS OF BOUNDARY MASK ON CHOLEC80 DATASET.

Mask width	AC (%)	PR (%)	RE (%)	JA (%)
0	91.14	90.84	87.64	79.14
3	92.04	91.87	89.07	81.44
5	92.31	92.26	89.52	82.12
10	**92.75**	92.86	**90.57**	**83.32**
20	92.68	**93.20**	90.40	82.66

To further explain this phenomenon, we visualize the feature similarity matrix in Fig. 12. Each red box in each similarity matrix indicates each phase in a video. It is clear that training with our generate pseudo labels *i.e.*, removing ambiguous labels near boundaries between two phases, would help to decrease intra-class distance and increase inter-class distance simultaneously.

V. DISCUSSION

Surgical phase recognition is one key component of computer-assisted surgery systems, which advances context awareness in modern operating rooms. However, most existing works require full annotations which are expensive, expertise-required and error-prone [31]. In contrast, we introduce timestamp supervision which only requires one timestamp annotated by humans for each phase in a video. We invite two surgeons to conduct both full and timestamp annotations and record the time cost for these two annotations. To leverage this supervision, we propose **U**ncertainty-**A**ware **T**emporal **D**iffusion (**UATD**) to generate trustworthy pseudo labels for those unlabeled frames, which is based on the property of surgical phases. Furthermore, loop training is also introduced to address the imbalance training and memory cost in timestamp surgical phase recognition. The in-depth empirical studies of the proposed UATD and LP based on timestamp supervision discover four deep insights: **1)** Timestamp annotation can reduce 74% annotation time compared with the full annotation, and surgeons tend to annotate those timestamps that are near the middle of phases; **2)** Extensive experiments demonstrate that our method can achieve competitive results compared with full supervision methods while reducing manual annotation costs; **3)** Less is more in surgical phase recognition, *i.e.*, less but discriminative pseudo labels outperform full but containing ambiguous frames; **4)** The proposed UATD can be used as a plug-and-play method to clean ambiguous labels near boundaries between phases, and improve the performance of the current surgical phase recognition methods; see details in Table IX.



Training with GT

Training with Ours

Fig. 12. Feature similarity matrix visualization. The horizontal and vertical axes represent the time indexes. We use cosine similarity to measure the degree of similarity between two arbitrary frame-level feature vectors within the same video. Each red box indicates each phase in a video. Note that, frame-level features of the same phase should be as similar as possible while separating one from others. Compared to the model trained with the ground-truth (GT), better representations of the features can be learned by our generated pseudo labels (right).

Although our method achieves promising results, there are some limitations. First, the temporal property we consider is not overall yet. The diffusion in our method assumes that the workflow is smooth without dramatic change and hardly any ambiguous frame occurs in the internal phase ; see Fig. 4 (e)-(f). But such an assumption may be false for other datasets and in the future, we will study more comprehensive temporal relationships to handle the intra-phase discontinuity. Moreover, the training process we propose is time-consuming containing several iterations of the training model from scratch. And we will design a more elegant training process to link up the optimal learning from different annotations, *i.e.*, different rounds of temporal diffusion in our methods.

Finally, we expect the community to focus more on label-efficient surgical video analysis. The weakly setting of videos, such as transcripts [42] and timestamp supervision, deserve further attention and exploitation. And the related ideas can be further investigated in other medical image analysis problems in CT [65]–[68], MRI [69]–[71].

VI. Conclusion

In this paper, we introduce the most annotation-saving setting, namely timestamp supervision, for surgical phase recognition. With timestamp supervision, we propose a novel uncertainty-aware temporal diffusion (UATD) method to generate trustworthy pseudo labels according to the labeled frames. Our main idea is to generate pseudo labels by considering the relationship among video frames. Results on two datasets show that our method can achieve competitive performance compared with the fully supervised setup. Moreover, we also find that our method can be used as a labeling clean approach to remove the noisy labels near boundaries to improve the generalization of the current surgical phase recognition, which reveals an interesting phenomenon less is more in this task. This paper provides some insights for label-efficient surgical phase recognition and hopefully inspires researchers to design label-efficient surgical video analysis algorithms.

References

[1] L. Maier-Hein, S. S. Vedula, S. Speidel, N. Navab, R. Kikinis, A. Park, M. Eisenmann, H. Feussner, G. Forestier, S. Giannarou, *et al.*, "Surgical data science for next-generation interventions," *Nature Biomedical Engineering*, vol. 1, no. 9, pp. 691–696, 2017.

[2] A. Moglia, V. Ferrari, L. Morelli, M. Ferrari, F. Mosca, and A. Cuschieri, "A systematic review of virtual reality simulators for robot-assisted surgery," *European urology*, vol. 69, no. 6, pp. 1065–1080, 2016.

[3] Y. Jin, Q. Dou, H. Chen, L. Yu, J. Qin, C.-W. Fu, and P.-A. Heng, "Sv-rcnet: workflow recognition from surgical videos using recurrent convolutional network," *IEEE transactions on medical imaging*, vol. 37, no. 5, pp. 1114–1126, 2017.

[4] Y. Jin, Y. Long, C. Chen, Z. Zhao, Q. Dou, and P.-A. Heng, "Temporal memory relation network for workflow recognition from surgical video," *IEEE Transactions on Medical Imaging*, 2021.

[5] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, "Endonet: a deep architecture for recognition tasks on laparoscopic videos," *IEEE transactions on medical imaging*, vol. 36, no. 1, pp. 86–97, 2016.

[6] N. Bricon-Souf and C. R. Newman, "Context awareness in health care: A review," *international journal of medical informatics*, vol. 76, no. 1, pp. 2–12, 2007.

[7] B. Bhatia, T. Oates, Y. Xiao, and P. Hu, "Real-time identification of operating room state from video," in *AAAI*, vol. 2, pp. 1761–1766, 2007.

[8] D. Liu, Q. Li, T. Jiang, Y. Wang, R. Miao, F. Shan, and Z. Li, "Towards unified surgical skill assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9522–9531, 2021.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[12] X. Ding and X. Li, "Exploring segment-level semantics for online phase recognition from surgical videos," *IEEE Transactions on Medical Imaging*, 2022.

[13] X. Gao, Y. Jin, Y. Long, Q. Dou, and P.-A. Heng, "Trans-svnet: Accurate phase recognition from surgical videos via hybrid embedding aggregation transformer," *arXiv preprint arXiv:2103.09712*, 2021.

[14] Z. Li, Y. Abu Farha, and J. Gall, "Temporal action segmentation from timestamp supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8365–8374, 2021.

[15] Y. Jin, H. Li, Q. Dou, H. Chen, J. Qin, C.-W. Fu, and P.-A. Heng, "Multi-task recurrent convolutional network with correlation loss for surgical video analysis," *Medical image analysis*, vol. 59, p. 101572, 2020.

[16] T. Blum, H. Feußner, and N. Navab, "Modeling and segmentation of surgical workflow from laparoscopic video," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 400–407, Springer, 2010.

[17] F. Lalys, L. Riffaud, D. Bouget, and P. Jannin, "A framework for the recognition of high-level surgical tasks from video images for cataract surgeries," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 966–976, 2011.

[18] A. Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, 2011.

[19] I. Funke, S. Bodenstedt, F. Oehme, F. von Bechtolsheim, J. Weitz, and S. Speidel, "Using 3d convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 467–475, Springer, 2019.

[20] F. Yi and T. Jiang, "Hard frame detection and online mapping for surgical phase recognition," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 449–457, Springer, 2019.

[21] F. Yi and T. Jiang, "Not end-to-end: Explore multi-stage architecture for online surgical phase recognition," *arXiv preprint arXiv:2107.04810*, 2021.

[22] B. Zhang, A. Ghanem, A. Simes, H. Choi, A. Yoo, and A. Min, "Swnet: Surgical workflow recognition with deep convolutional network," in *Medical Imaging with Deep Learning*, 2021.

[23] X. Ding, Z. Liu, and X. Li, "Free lunch for surgical video understanding by distilling self-supervisions," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, (Cham), pp. 365–375, Springer Nature Switzerland, 2022.

[24] M. Sahu, A. Szengel, A. Mukhopadhyay, and S. Zachow, "Surgical phase recognition by learning phase transitions," *Current Directions in Biomedical Engineering*, vol. 6, no. 1, 2020.

[25] T. Czempiel, M. Paschali, M. Keicher, W. Simson, H. Feussner, S. T. Kim, and N. Navab, "Tecno: Surgical phase recognition with multi-stage temporal convolutional networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 343–352, Springer, 2020.

[26] T. Czempiel, M. Paschali, D. Ostler, S. T. Kim, B. Busam, and N. Navab, "Opera: Attention-regularized transformers for surgical phase recognition," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 604–614, Springer, 2021.

[27] W. Dai, X. Li, X. Ding, and K.-T. Cheng, "Cyclical self-supervision for semi-supervised ejection fraction prediction from echocardiogram videos," *IEEE Transactions on Medical Imaging*, 2022.

[28] H. Yao, X. Hu, and X. Li, "Enhancing pseudo label quality for semi-supervised domain-generalized medical image segmentation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 3099–3107, Jun. 2022.

[29] T. Yu, D. Mutter, J. Marescaux, and N. Padoy, "Learning from a tiny dataset of manual annotations: a teacher/student approach for surgical phase recognition," *arXiv preprint arXiv:1812.00033*, 2018.

[30] G. Yengera, D. Mutter, J. Marescaux, and N. Padoy, "Less is more: Surgical phase recognition with less annotations through self-supervised pre-training of cnn-lstm networks," *arXiv preprint arXiv:1805.08569*, 2018.

[31] R. DiPietro and G. D. Hager, "Automated surgical activity recognition with one labeled sequence," in *International conference on medical image computing and computer-assisted intervention*, pp. 458–466, Springer, 2019.

[32] X. Shi, Y. Jin, Q. Dou, and P.-A. Heng, "Lrtd: long-range temporal dependency based active learning for surgical workflow recognition," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 9, pp. 1573–1584, 2020.

[33] X. Shi, Y. Jin, Q. Dou, and P.-A. Heng, "Semi-supervised learning with progressive unlabeled data excavation for label-efficient surgical workflow recognition," *Medical Image Analysis*, vol. 73, p. 102158, 2021.

[34] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *2017 IEEE international conference on computer vision (ICCV)*, pp. 3544–3553, IEEE, 2017.

[35] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 4325–4334, 2017.

[36] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in *Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6752–6761, 2018.

[37] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 563–579, 2018.

[38] X. Ding, N. Wang, J. Li, and X. Gao, "Weakly supervised temporal action localization with segment-level labels," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 42–54, Springer, 2021.

[39] X. Ding, N. Wang, X. Gao, J. Li, X. Wang, and T. Liu, "Kfc: An efficient framework for semi-supervised temporal action localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 6869–6878, 2021.

[40] X. Ding, N. Wang, S. Zhang, D. Cheng, X. Li, Z. Huang, M. Tang, and X. Gao, "Support-set based cross-supervision for video grounding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11573–11582, 2021.

[41] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic, "Weakly supervised action labeling in videos under ordering constraints," in *European Conference on Computer Vision*, pp. 628–643, Springer, 2014.

[42] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling," in *European Conference on Computer Vision*, pp. 137–153, Springer, 2016.

[43] J. Li, P. Lei, and S. Todorovic, "Weakly supervised energy-based learning for action segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6243–6251, 2019.

[44] A. Richard, H. Kuehne, and J. Gall, "Action sets: Weakly supervised action segmentation without ordering constraints," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5987–5996, 2018.

[45] P. Mettes, J. C. Van Gemert, and C. G. Snoek, "Spot on: Action localization from pointly-supervised proposals," in *European conference on computer vision*, pp. 437–453, Springer, 2016.

[46] D. Moltisanti, S. Fidler, and D. Damen, "Action recognition from single timestamp supervision in untrimmed videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9915–9924, 2019.

[47] F. Ma, L. Zhu, Y. Yang, S. Zha, G. Kundu, M. Feiszli, and Z. Shou, "Sf-net: Single-frame supervision for temporal action localization," in *European conference on computer vision*, pp. 420–437, Springer, 2020.

[48] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *arXiv preprint arXiv:1703.04977*, 2017.

[49] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, PMLR, 2016.

[50] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[51] R. Tanno, D. E. Worrall, A. Ghosh, E. Kaden, S. N. Sotiropoulos, A. Criminisi, and D. C. Alexander, "Bayesian image quality transfer with cnns: exploring uncertainty in dmri super-resolution," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 611–619, Springer, 2017.

[52] A. Jungo, F. Balsiger, and M. Reyes, "Analyzing the quality and challenges of uncertainty estimations for brain tumor segmentation," *Frontiers in neuroscience*, p. 282, 2020.

[53] Y. Lin, H. Yao, Z. Li, G. Zheng, and X. Li, "Calibrating label distribution for class-imbalanced barely-supervised knee segmentation," *arXiv preprint arXiv:2205.03644*, 2022.

[54] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[55] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[56] M.-H. Laves, S. Ihler, and T. Ortmaier, "Uncertainty quantification in computer-aided diagnosis: Make your model say" i don't know" for ambiguous cases," *arXiv preprint arXiv:1908.00792*, 2019.

[57] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Scientific reports*, vol. 7, no. 1, pp. 1–14, 2017.

[58] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, 2019.

[59] Y. A. Farha and J. Gall, "Ms-tcn: Multi-stage temporal convolutional network for action segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3575–3584, 2019.

[60] A. P. Twinanda, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "Single-and multi-task architectures for surgical workflow challenge at m2cai 2016," *arXiv preprint arXiv:1610.08844*, 2016.

[61] A. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, "Miccai modeling and monitoring of computer assisted interventions challenge," 2016.

[62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[64] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2025–2041, 2017.

[65] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, "H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes," *IEEE transactions on medical imaging*, vol. 37, no. 12, pp. 2663–2674, 2018.

[66] E. Gibson, F. Giganti, Y. Hu, E. Bonmati, S. Bandula, K. Gurusamy, B. Davidson, S. P. Pereira, M. J. Clarkson, and D. C. Barratt, "Automatic multi-organ segmentation on abdominal ct with dense v-networks," *IEEE transactions on medical imaging*, vol. 37, no. 8, pp. 1822–1834, 2018.

[67] T. Heimann, B. Van Ginneken, M. A. Styner, Y. Arzhaeva, V. Aurich, C. Bauer, A. Beck, C. Becker, R. Beichel, G. Bekes, *et al.*, "Comparison and evaluation of methods for liver segmentation from ct datasets," *IEEE transactions on medical imaging*, vol. 28, no. 8, pp. 1251–1265, 2009.

[68] X. Li, L. Yu, H. Chen, C.-W. Fu, L. Xing, and P.-A. Heng, "Transformation-consistent self-ensembling model for semisupervised medical image segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 523–534, 2020.

[69] T. Wang, X. Xu, J. Xiong, Q. Jia, H. Yuan, M. Huang, J. Zhuang, and Y. Shi, "Ica-unet: Ica inspired statistical unet for real-time 3d cardiac cine mri segmentation," in *International conference on medical image computing and computer-assisted intervention*, pp. 447–457, Springer, 2020.

[70] X. Li, Q. Dou, H. Chen, C.-W. Fu, X. Qi, D. L. Belavỳ, G. Armbrecht, D. Felsenberg, G. Zheng, and P.-A. Heng, "3d multi-scale fcn with random modality voxel dropout learning for intervertebral disc localization and segmentation from multi-modality mr images," *Medical image analysis*, vol. 45, pp. 41–54, 2018.

[71] Y. Yu, Y. Xie, T. Thamm, E. Gong, J. Ouyang, C. Huang, S. Christensen, M. P. Marks, M. G. Lansberg, G. W. Albers, *et al.*, "Use of deep learning to predict final ischemic stroke lesions from initial magnetic resonance imaging," *JAMA network open*, vol. 3, no. 3, pp. e200772–e200772, 2020.