Medical Image Analysis 101 (2025) 103471

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

SAF-IS: A spatial annotation free framework for instance segmentation of surgical tools

Luca Sestini [a,b,*], Benoit Rosa [a], Elena De Momi [b], Giancarlo Ferrigno [b], Nicolas Padoy [a,c]

[a] ICube, University of Strasbourg, CNRS, France
[b] Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milano, Italy
[c] IHU Strasbourg, Strasbourg, France

ARTICLE INFO

Keywords:
Endoscopic videos
Instance segmentation
Tool segmentation
Weakly supervised learning
Self supervised learning

ABSTRACT

Instance segmentation of surgical instruments is a long-standing research problem, crucial for the development of many applications for computer-assisted surgery. This problem is commonly tackled via fully-supervised training of deep learning models, requiring expensive pixel-level annotations to train.

In this work, we develop a framework for instance segmentation not relying on spatial annotations for training. Instead, our solution only requires binary tool masks, obtainable using recent unsupervised approaches, and tool presence labels, freely obtainable in robot-assisted surgery. Based on the binary mask information, our solution learns to extract individual tool instances from single frames, and to encode each instance into a compact vector representation, capturing its semantic features. Such representations guide the automatic selection of a tiny number of instances (8 only in our experiments), displayed to a human operator for tool-type labelling. The gathered information is finally used to match each training instance with a tool presence label, providing an effective supervision signal to train a tool instance classifier.

We validate our framework on the EndoVis 2017 and 2018 segmentation datasets. We provide results using binary masks obtained either by manual annotation or as predictions of an unsupervised binary segmentation model. The latter solution yields an instance segmentation approach completely free from spatial annotations, outperforming several state-of-the-art fully-supervised segmentation approaches.

1. Introduction

Endoscopic videos from minimally invasive procedures offer rich information describing the surgical act. The automatic analysis of such information opens up several opportunities to better understand the surgical practice and to improve it (Francis et al., 2018; Lavanchy et al., 2021; Mascagni et al., 2022). Surgical computer vision provides the necessary tools to process raw endoscopic videos, enabling the extraction of dense information for downstream applications. Among the various surgical computer vision tasks, automatic instrument localization and identification represent an essential component of many downstream applications, like surgical skill assessment (Lavanchy et al., 2021), augmented reality (Tanzi et al., 2021), 3D scene reconstruction (Wang et al., 2022) and 3D pose estimation (Allan et al., 2018). This problem is often formalized by means of either *semantic* or *instance* segmentation. Semantic Segmentation (SeS) aims at directly labelling each image pixel as either belonging to the *background* class or to a certain *tool type* class. Instance Segmentation (IS) aims at localizing and identifying individual tool instances, providing, for each instance, a segmentation mask and a tool-type label. Such *tool instantiation* information, *i.e.* the availability of a separate segmentation mask for each tool instance present in the image, is extremely precious for downstream applications like automatic skill assessment, as it enables individual tool tracking over time. State-of-the-art approaches commonly tackle tool segmentation via fully-supervised training of deep learning models (Shvets et al., 2018; Jin et al., 2019; Kong et al., 2021; Kurmann et al., 2021). Such approaches require the availability of pixel-level semantic and instance labels, extremely expensive to collect via manual annotation at a large scale. This confines the training of such models to small annotated datasets, limiting their generalization ability.

Recently, alternatives to standard fully-supervised approaches have been proposed for the task of binary tool segmentation, a type of SeS featuring only two classes, *tool* and *background* (Sahu et al., 2020; Sestini et al., 2023; Pakhomov et al., 2020; Sestini et al., 2021; da Costa Rocha et al., 2019). Most of these solutions rely on semi-synthetic dataset generation, for example by combining simulation

* Corresponding author at: ICube, University of Strasbourg, CNRS, France.
E-mail address: sestini@unistra.fr (L. Sestini).

https://doi.org/10.1016/j.media.2025.103471
Received 5 August 2023; Received in revised form 7 November 2024; Accepted 11 January 2025
Available online 22 January 2025

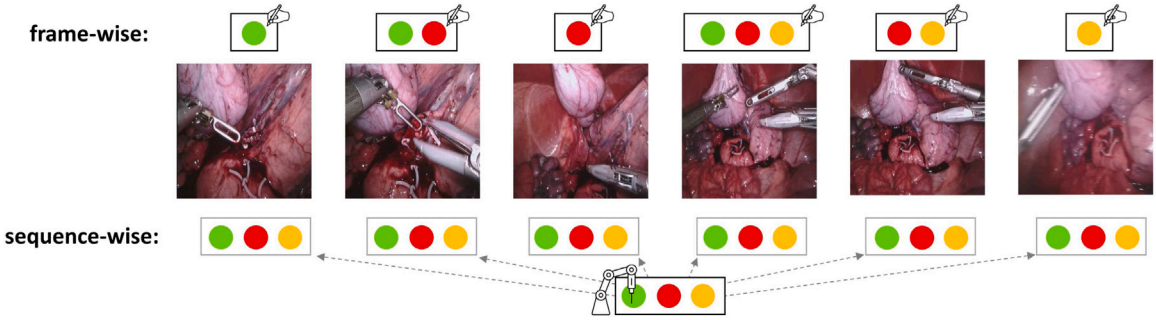

Fig. 1. Examples of *frame-wise* and *sequence-wise* tool presence labels for a robot-assisted surgery sequence from the EndoVis 2017 dataset (each colour represents a tool type). Given a sequence of frames, the *sequence-wise* label obtained from the robotic system is assigned to each frame. Notice how all the tools can be attached to the system at the same time, while being visible only in certain frames.

data and domain translation approaches (Sahu et al., 2020). While appealing, their application is still potentially limited by the domain gap between synthetic and real data, and by the need for ad-hoc setups to collect synthetic data. A few alternative works have shown the potential of prior instrument knowledge to train deep learning models for binary segmentation, without requiring spatial annotations. This has been done by exploiting prior knowledge on instrument motion and shape (Sestini et al., 2023), or, in the robotic context, by incorporating 3D tool models and kinematic data (da Costa Rocha et al., 2019; Pakhomov et al., 2020; Sestini et al., 2021; Ding et al., 2022).

Despite these growing efforts to reduce the dependency on manual annotations, research has remained confined to the binary segmentation task. We believe that this is due to the rigid problem formalization imposed by common instance and semantic segmentation approaches: such approaches do not benefit from the potential availability of binary segmentation masks, as they would still require pixel-level labels to train. Furthermore, this problem formalization prevents the incorporation of significantly cheaper sources of semantic information, compared to spatial annotations, like tool presence labels. Specifically, we define as *frame-wise* the tool presence labels indicating which tool types are *effectively* visible in each frame; we define as *sequence-wise* the labels indicating which tool types are *potentially* visible in each frame (see Fig. 1 for examples of a sequence). While *frame-wise* labels are usually obtained via manual annotation – although much cheaper than spatial annotation – *sequence-wise* labels can be automatically obtained from different sources. In robot-assisted surgery, for example, robotic systems can often record which tools are attached (Kurmann et al., 2021). This information only indicates that a certain tool could be visible at some point while it is attached, but does not guarantee its visibility in any specific frame (therefore a *sequence-wise* visibility). As a generalization, surgical phase and step annotations could provide similar information, when a mapping between phases/steps and tools can be approximately defined, for example by knowing which tools are commonly used in each phase/step (Padoy et al., 2012). While tool presence labels have been extensively explored for tool localization via bounding-box detection (Vardazaryan et al., 2018; Nwoye et al., 2019; Zia et al., 2023), their use for the segmentation task is still extremely limited.

In this work we propose a framework for instance segmentation model training, which embraces the recent progress on unsupervised binary segmentation and the availability of cheap tool presence labels, either *frame-wise* or *sequence-wise*. Compared to pixel-level annotations, tool presence labels are not spatially localized. Weakly-supervised tool detection approaches (Vardazaryan et al., 2018; Nwoye et al., 2019; Zia et al., 2023) often exploit the class activation maps provided by a classifier trained on the tool presence labels, in order to localize the tools in the image space. However, such localization is commonly limited to discriminant parts of the tools, like the tip, and thus not suitable for segmentation. In addition, such approaches struggle to handle *sequence-wise* labels, as these labels do not provide a ground truth signal for the training of the frame-wise classifier. To tackle these challenges our solution first learns to localize individual tool instances and to encode each of them in a compact feature representation. These instance-wise representations are then used to automatically select a small number of tool instances (*prototype instances*, as few as 8 in our experiments), which are presented to a human operator for tool-type labelling. The gathered information is finally used to match each instance to a tool-type label from the corresponding set of tool presence labels, providing an effective supervision signal for the training of an instance classifier.

To this aim, we make the following contributions:

- we develop an unsupervised approach for tool instantiation (Fig. 2, *Tool instantiation*). This step allows training a model to extract a separate binary segmentation mask for each tool instance present in a frame. With no availability of pixel-level labels, we fabricate a pseudo-supervision signal from the connected component instantiation of the binary masks, and refine it using simple assumptions on instrument positioning in the image space. This signal is used to train the instantiation model, directly predicting the position of each instance centroid in the image space in the form of a 2D displacement field;
- we develop a self-supervised approach for feature representation learning (Fig. 2, *Feature representation learning*): this step allows training a model to encode each tool instance in a compact representation, capturing its semantic features. With no availability of pixel-level semantic labels, we learn such representations by relying on intrinsic temporal information from video sequences. Specifically, we design a contrastive learning approach based on local instance tracking to draw positive and negative samples. This step allows obtaining powerful instance-wise feature representations, providing the necessary information to solve the final classification training step;
- we develop an approach to learn instance classification from the tool presence labels (Fig. 2, *Instance classification*). The feature representations of all the training instances, learnt in the previous step, are used to guide the automatic selection of a tiny number of prototype instances, displayed to a human operator for tool-type labelling. The gathered information is propagated to the whole training set, allowing us to label each training instance with a pseudo tool-type label (*prototype labels*). This information is combined with the available tool presence labels (either *frame-wise* or *sequence-wise*) using a Teacher-Student approach. This step allows matching each training instance to a tool-type label from the corresponding set of tool presence labels, providing an effective supervision signal for the training of the student instance classifier;
- at inference time the trained architecture can perform instance segmentation on single frames, by extracting individual tool instances, encoding each of them in a compact feature representation, and separately classifying them (Fig. 2, bottom).

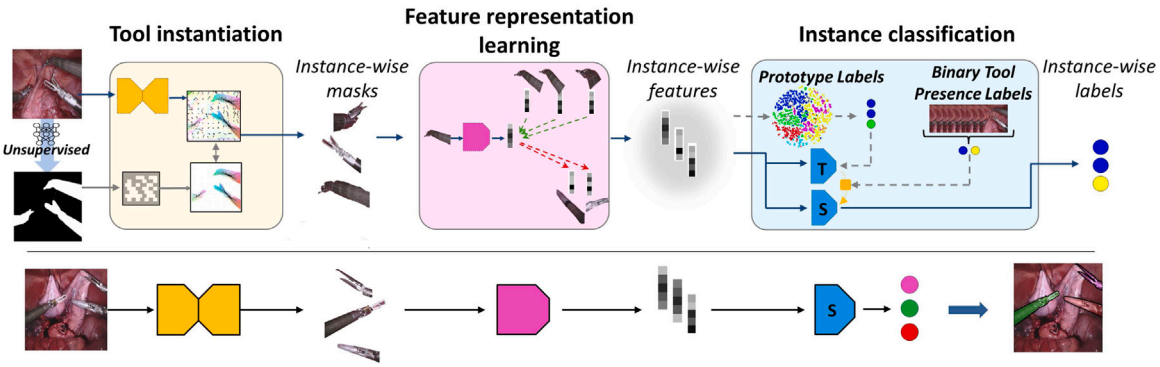

Fig. 2. Overview of the proposed Spatial Annotation Free framework for Instance Segmentation (SAF-IS). Top: training architecture highlighting the three core steps. *Tool instantiation* is learnt from binary masks, potentially obtained using existing unsupervised segmentation methods. *Feature representation learning* is performed using a contrastive learning strategy, powered by local temporal tracking. This step allows us to extract a feature representation of each tool instance in the training set. *Instance classification* is performed by incorporating a minimal amount of human-provided information (*prototype labels*, as few as 8 in our experiments) and cheaply obtainable tool presence labels. Bottom: SAF-IS inference architecture.

2. Related work

Surgical instrument segmentation is a long-standing research problem. Before the Deep Learning (DL) breakthrough, the problem was tackled by totally relying on prior knowledge about surgical tools, like colour distribution (Wei et al., 1997), shape (Bouget et al., 2015) or orientation in the field-of-view (Voros et al., 2006). Following the DL breakthrough in the computer vision field, a great research effort has been dedicated to designing powerful fully-supervised architectures, boosting segmentation accuracy. Such solutions are presented below in Section 2.1, focusing in particular on instance segmentation approaches. Although fully-supervised methods have achieved unprecedented segmentation results on benchmark datasets, their scalability is restricted by the need for manual annotations, which confines their training to small annotated datasets, limiting their generalization ability. To address this challenge, various approaches have been suggested, which we present in Section 2.2.

2.1. Fully supervised solutions

Following the DL breakthrough in the field of surgical computer vision, research works have mostly addressed the problem of surgical tool segmentation using fully-supervised DL approaches. In particular, encoder–decoder architectures based on Convolutional Neural Networks (CNNs) have been widely adopted, in concurrency with a semantic segmentation formulation of the problem. Garcia-Peraza-Herrera et al. (2017), Shvets et al. (2018), Pakhomov et al. (2019) and Hasan and Linte (2019) propose different variations of the U-Net architecture (Ronneberger et al., 2015), exploring different loss functions, residual connections, dilated convolutions and ad-hoc augmentation pipelines. Multi-task learning has also been adopted, coupling the segmentation task with image-based localization of tool landmarks (Laina et al., 2017) and task-oriented saliency maps prediction (Islam et al., 2021). While the segmentation task can be solved for single frames, temporal information, for example in the form of optical flow, has been shown to boost performance (Jin et al., 2019).

Recently, instance segmentation approaches have started gaining traction. Several of the proposed approaches are based on the popular Mask-RCNN architecture (He et al., 2017). Kong et al. (2021) directly train a Mask-RCNN architecture for the task of surgical instrument instance segmentation. ISI-Net (González et al., 2020) adds a temporal consistency module for improved segmentation results. Kurmann et al. (2021) propose a *mask-then-classify* approach, adopting an anchor-free approach for instrument instantiation, based on direct localization of instruments centroids. Differently from the above-listed methods, Zhao et al. (2022) simultaneously tackle the problems of instance segmentation and tracking using a transformer architecture based on the popular TrackFormer and DETR models (Carion et al., 2020; Meinhardt et al., 2022).

In this work we also adopt an instance segmentation problem formalization, showing its benefits beyond fully-supervised training.

2.2. Non fully-supervised solutions

Motivated by the need to reduce the burden of manual annotation, several solutions have tackled the segmentation problem by including unlabelled data in the training process, exploiting small sets of labelled data, weak annotations or prior knowledge. Such solutions, mostly focusing on the binary segmentation problem, are presented below.

Semi-Supervised solutions: this family of approaches incorporates unlabelled data in the training process, while still requiring access to a set of manually annotated data. Different solutions to combine unlabelled and labelled data have been explored. Ross et al. (2018) pre-train a CNN on unlabelled data, by means of a pretext task carried out using a cycle-GAN architecture, and then fine-tune the model on annotated data. A similar pipeline can be followed by replacing the pre-text task with self-supervised representation learning on the unlabelled data, as experimentally shown by Ramesh et al. (2022). Zhao et al. (2020) tackle the problem of sparsely annotated data, propagating low hertz annotations to intermediate unlabelled frames using optical flow. Kalia et al. (2021) incorporate unlabelled data from different domains in the training process to improve generalization to these domains. This is achieved by mapping annotated frames from the labelled set to the unlabelled domain using a cycle-GAN architecture, allowing for better generalization.

Weakly-Supervised solutions: weakly-supervised learning approaches aim at training machine learning models using annotations cheaper to obtain compared to the ones required by fully-supervised solutions. Such annotations could be a simplification of the ideal ground truth annotations, like scribbles in place of masks for the segmentation task, *weak* annotations providing indirect supervision for the target task (e.g. tool presence labels to solve tool localization tasks), or automatically acquired multi-modal data.

The use of weakly-supervised approaches to train tool segmentation models is mostly confined to the binary segmentation task. Lee et al. (2019) propose a framework to integrate scribble-like annotations, speeding up the annotation process. Multi-modal data have also been largely explored in different forms. Yang et al. (2022) automatically obtain a pseudo-supervision signal by attaching an electromagnetic sensor to the surgical instruments. While cutting the cost of annotations, the approach is inherently limited by regulatory constraints, which limit the extent of validation of this study. Alternatively, kinematic data, in combination with tool-specific kinematic models, can provide

an effective source of supervision for robotic tool segmentation. Ding et al. (2022) propose CaRTS, a framework describing the causal relationship between observed kinematics and corresponding endoscopic image, allowing to refine the first based on the information provided by the latter for accurate and robust binary tool segmentation. In a follow-up publication, Ding et al. (2023) CaRTS framework was extended, decoupling time-variant and time-invariant factors determining instrument configuration. This allows to separately optimize kinematic values, camera-robot transformation and camera parameters, yielding improved segmentation results and reduced inference time. da Costa Rocha et al. (2019) combine kinematic joint values and 3D kinematic models of flexible surgical instruments to generate pseudo ground truth segmentation masks, subsequently refined using a GrabCut segmentation algorithm. Similarly, Pakhomov et al. (2020) generate pseudo segmentation masks from the recorded kinematics and refine them using a cycle-GAN architecture. The application of such approaches is currently limited to the domain of robot-assisted surgery, as kinematic data are not available for manual laparoscopy. In addition, these approaches require the availability of precise kinematic models of the tools, not always available in practice.

Weak annotations, in the form of *frame-wise* tool presence labels, have been used to tackle the problem of tool localization via bounding-box detection. Vardazaryan et al. (2018) train a multi-label classifier to predict tool presence from single frames; the designed architecture features an extended spatial pooling layer yielding class-specific feature maps, the *class activation maps*, used during inference to localize the tools. Similarly (Nwoye et al., 2019) use Wildcat Pooling (Durand et al., 2017) to obtain localization maps, adding a convolution-LSTM module for improved temporal consistency. Differently from these two approaches, Xue et al. (2022) use tool presence labels, in combination with green-screen recorded images of surgical instruments, to obtain a pseudo-supervision signal consisting of noisy and redundant bounding boxes. A bounding-box regressor is then trained on the noisy supervision signal, and its predictions for a certain tool are averaged together according to their confidence score. Recently, a challenge entirely dedicated to the problem of weakly-supervised tool localization, named SurgToolLoc, has been held at EndoVis 2022 (Zia et al., 2023). The aim of the challenge was to leverage *sequence-wise* tool presence labels in order to train machine learning models to detect and localize tools in endoscopic frames. Specifically, the problem was formalized as bounding-box detection of the tool clevis. The eight submitting teams proposed various approaches to combine class activation maps and transfer learning from publicly available fully-annotated datasets. Among the submitting teams, only three obtained a significant performance (mean average precision $> 10\%$), a clear indication of the challenges that this problem poses.

Despite these efforts, the use of tool presence labels has remained limited to the bounding-box detection task, as the standard approach involving using class activation maps limits the localization to discriminative parts of the tools, missing out significant parts of the instruments like the shaft.

Prior knowledge based solutions: as shown by early works on tool segmentation, general assumptions on colour distribution of endoscopic frames, instrument position and prior shape knowledge, can be a sufficient source of information to localize surgical instruments. Liu et al. (2020), for example, generate segmentation pseudo-labels using handcrafted cues, such as colour distribution; binary segmentation results are then refined by exploiting feature correlation between adjacent video frames. Sestini et al. (2023) propose FUN-SIS, an approach exploiting general assumptions on instrument motion and *shape-priors* to train a binary segmentation model, achieving results comparable to the ones of fully-supervised solutions.

In this work we combine the use of prior knowledge and tool presence labels to learn instance segmentation of surgical instruments. Prior knowledge on instrument positioning in the field-of-view is exploited to instantiate binary segmentation masks. Weak information, in the form of tool presence labels, both *frame-wise* and *sequence-wise*, is then incorporated to achieve accurate instance classification.

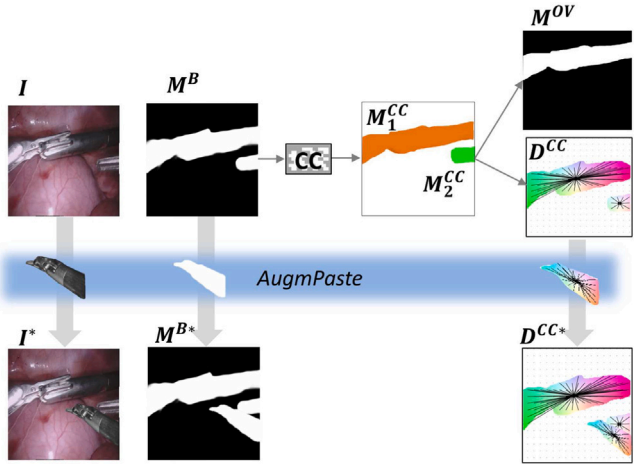

Fig. 3. Overview of the proposed strategy to generate a pseudo-supervision signal to learn instrument instantiation. Given an image I, its binary mask M^B is instantiated using a Connected Component (CC) algorithm, yielding the set of tool masks $\{M_i^{CC}\}$, with i in $[1, N_{CC}]$. From them, the displacement field D^{CC} and the overlap mask M^{OV} can be automatically obtained. A random tool instance is then selected from the training set, and pasted on I, M^B, D^{CC}, producing their augmented versions I^*, M^{B*}, D^{CC*}.

3. Methodology

The proposed SAF-IS framework for Spatial Annotation Free Instance Segmentation explicitly separates the task into three core components: tool instantiation, feature representation and instance classification. Differently from standard semantic/instance segmentation approaches, SAF-IS does not require spatial annotation of the training data. Instead, it relies on the availability of binary segmentation masks, which can be cheaply obtained using emerging unsupervised approaches, and tool presence labels.

The full framework is presented in Fig. 2 and detailed below.

3.1. Tool instantiation

Instrument instantiation is here defined as the problem of predicting, from an endoscopic image I, the set of binary masks $\{M_i^{Inst}\}$, with i in $[1, N_{Inst}]$, each one corresponding to an individual instrument visible in the image. When the ground truth instantiation is known, the problem is often formulated as bounding-box prediction (Kong et al., 2021; González et al., 2020). However, the effectiveness of this approach has been questioned in Kurmann et al. (2021), which proposed an alternative solution based on direct regression of instance centroids' position. We here adopt a similar formulation, showing its benefits with respect to bounding-box prediction beyond fully-supervised learning.

The instantiation problem is here formalized as learning the mapping between the image $I \in R^{W \times H \times 3}$ and the displacement field $D \in R^{W \times H \times 2}$, uniquely assigning each tool pixel to an instance. Given a pixel $\mathbf{p} = [p_x, p_y]$, $D|_{\mathbf{p}}$ is equal to the vector $\mathbf{v} = [c_x^i - p_x, c_y^i - p_y]$ if $\mathbf{p}$ belongs to a certain instance i, having its centroid in $[c_x^i, c_y^i]$, or to the null vector $[0, 0]$, if $\mathbf{p}$ belongs to the background. Whenever the ground truth instantiation D is known, as in Kurmann et al. (2021), such mapping can be learnt by an instantiation model, implemented as a neural network, by using a fully-supervised training formulation. This can be achieved by optimizing the loss L_I^{FS}, implemented as the pixel-wise distance between the ground truth displacement field D and the instantiation model prediction $\tilde{D}$:

$$L_I^{FS} = |D - \tilde{D}|. \tag{1}$$

At inference time, given a new image I and the corresponding predicted displacement field $\tilde{D}$, the set of instance masks $\{M_i^{Inst}\}$ can

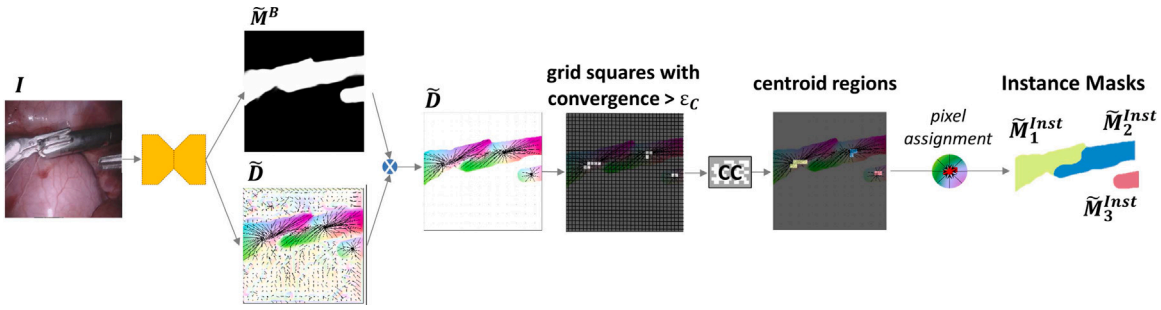

Fig. 4. Overview of the proposed instantiation strategy. Given an image I the trained instantiation model predicts the masked displacement field $\tilde{D}$. A square grid is then overlapped to $\tilde{D}$, and the squares with high convergence (per-pixel average > ε_C) are then extracted, and separated by Connected Component (CC) labelling, yielding a set of $\tilde{N}_{Inst}$ centroid regions. Each tool pixel is then assigned to the corresponding centroid, yielding the set of instance masks $\{\tilde{M}_i^{Inst}\}$, with i in $[1, \tilde{N}_{Inst}]$.

be easily extracted by identifying the instance centroids, as the pixels where the displacement field converges, and assigning each tool pixel to the centroid pointed by the corresponding displacement vector.

In our case, the ground truth displacement field is not known, and only the binary mask M^B is available.

Training: without a ground truth instantiation, we rely on the assumption that surgeons tend to avoid overlapping surgical instruments in the field-of-view, in order to reduce the chances of mutual tool occlusions and unwanted tool interactions. While generally true, this assumption does not always hold, especially during specific surgical activities, like suturing. For this reason, we relax this hypothesis when tools are likely to overlap, as detailed below.

Given an image I and the corresponding binary mask M^B, if tools do not overlap, the instance masks can be obtained by separating the Connected Components (CC) of M^B through standard computer vision methods like the Spaghetti algorithm (Bolelli et al., 2019). The displacement field D^{CC}, approximating the ground truth D, can then be directly obtained from the set of N_{CC} tool masks $\{M_i^{CC}\}$, with i in $[1, N_{CC}]$, by subtracting each tool pixel position from the centroid $[c_x^i, c_y^i]$ of the corresponding mask M_i^{CC}. While effective in the case of non-overlapping tools, CC labelling systematically fails when tools overlap. In order to mitigate this problem we artificially modify the supervision signal obtained from CC instantiation, as follows:

- **potential overlapping tools identification**: in minimally invasive surgery surgeons adopt the principle of *triangulation* to increase their ability to visualize and access anatomy (Russo, 2012). As a result, surgical tools commonly enter the camera's field-of-view from the sides. Therefore, given the set of CC masks $\{M_i^{CC}\}$, M_i^{CC} is considered a potential overlapping instance if it covers the full horizontal length of the frame (see Fig. 3 for an example). All pixels corresponding to potential overlapping instances are collected in the binary overlap mask M^{OV}, and discarded from loss computation as described later in this Section;

- **instance pasting augmentation (*AugmPaste*)**: given an image I, its binary mask M^B and its CC displacement field D^{CC}, a random tool instance is selected from a different training sample and pasted on them, yielding the augmented image I^*, the augmented binary mask M^{B*} and the augmented displacement field D^{CC*} (Fig. 3). This augmentation step allows us to artificially simulate the presence of overlapping instances, making up for the discarded instances in the previous step.

Given the image I^*, in addition to the displacement field $\tilde{D}$, we let the instantiation model predict the binary segmentation mask $\tilde{M}^B$, which we multiply by $\tilde{D}$ to ensure that the displacement vector for pixels belonging to the background is a null vector $[0, 0]$. For simplicity, we keep the notation $\tilde{D}$ to refer to the result of this product.

Given the image I^*, the corresponding network predictions $\tilde{D}$ and $\tilde{M}^B$, the binary mask M^{B*}, the displacement field D^{CC*} and the overlap mask M^{OV}, the instantiation model is trained by optimizing the loss L_I:

$$L_I = |D^{CC*} - \tilde{D}|(1 - M^{OV}) + L_{CE}(M^{B*}, \tilde{M}^B), \tag{2}$$

where L_{CE} is a standard pixel-wise cross-entropy loss.

Inference: given an image I and the trained instantiation model, the predicted displacement field $\tilde{D}$ must be mapped to the set of instance masks $\{\tilde{M}_i^{Inst}\}$, with i in $[1, \tilde{N}_{Inst}]$, and $\tilde{N}_{Inst}$ being the number of predicted instances in a frame. While for the ground truth displacement field D each tool pixel vector points exactly to the corresponding centroid pixel, this is not guaranteed for the predicted $\tilde{D}$. Therefore we define as *centroids* the regions of $\tilde{D}$ with a high rate of displacement vectors convergence. Practically, we overlap a square grid to $\tilde{D}$ and compute, for each square, the per-pixel average number of vectors pointing inside it. If such a number is above a predefined threshold ε_C, the square is considered a centroid square. Connected squares are grouped together, to yield the set of centroid regions $\{c_i\}$, with i in $[1, \tilde{N}_{Inst}]$. The instance masks can then be extracted by assigning each tool pixel $\mathbf{p}$ to the centroid c_i closest to the point identified by $\mathbf{p} + \tilde{D}|_{\mathbf{p}}$. This yields the set of predicted instance masks $\{\tilde{M}_i^{Inst}\}$, with i in $[1, \tilde{N}_{Inst}]$ (Fig. 4). In our framework, the predicted instance masks are subsequently used to learn instance-wise feature representations, as now discussed.

3.2. Feature representation learning

The aim of this step is learning to encode each tool instance in a compact feature representation, capturing its semantic features.

Training: in the absence of pixel-level semantic labels, we rely on self-supervision to learn robust and meaningful feature representations of each tool instance, tailored for the instance segmentation task. The problem of self-supervised representation learning has been often addressed by means of contrastive learning in literature (Jaiswal et al., 2020). While general contrastive learning approaches usually learn global frame-level feature representations, we find this formulation to be ill-posed for the instrument segmentation problem, as it lacks the spatial granularity necessary to discriminate between different instances. Therefore we design an instance-level contrastive learning approach, exploiting the unsupervised instantiation described above and intrinsic temporal information from video sequences.

Given an image I and the set of instance masks $\{\tilde{M}_i^{Inst}\}$ predicted by the instantiation model, we want to map each instance to a feature vector F_i, capturing its semantic content. We obtain feature vectors using a feature extractor model implemented using a standard ResNet-50 architecture. Specifically, for each instance, we pass I through the model and multiply the intermediate feature maps by $\tilde{M}_i^{Inst}$, resized to match their dimensions, to obtain the corresponding instance-wise feature vector F_i. Then, given a feature representation F_i, intrinsic temporal information from the video sequence is used to draw positive and negative examples for contrastive loss computation. Specifically:

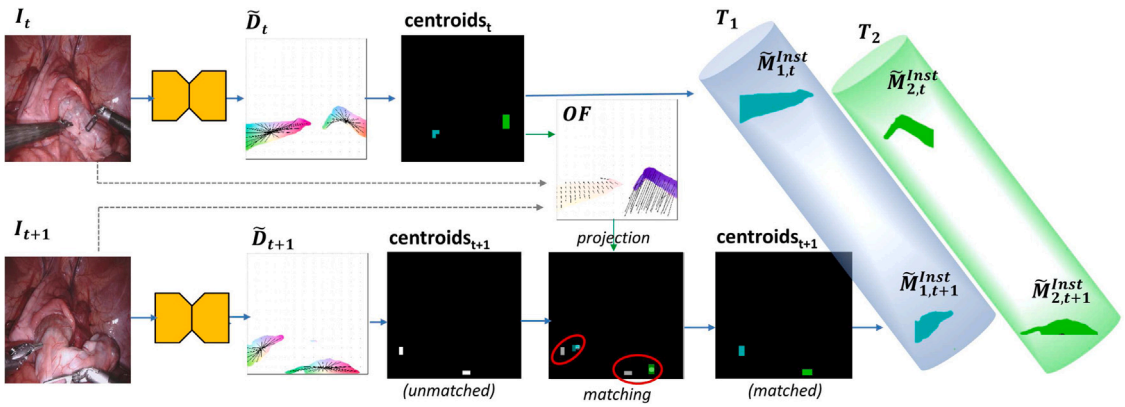

Fig. 5. Overview of the tracking strategy used to generate positive samples for contrastive learning. Given two consecutive frames, the centroids at time t, obtained from the displacement field D_t are mapped to the I_{t+1} space using optical flow OF, computed between the two images I_t and I_{t+1}. The projected centroids are then matched to the ones obtained from the displacement field D_{t+1}. This allows building the set of tubes $\{T_i\}$, with i in $[1, \tilde{N}_{Inst}]$. Tubes are progressively grown by repeating this process for consecutive frames.

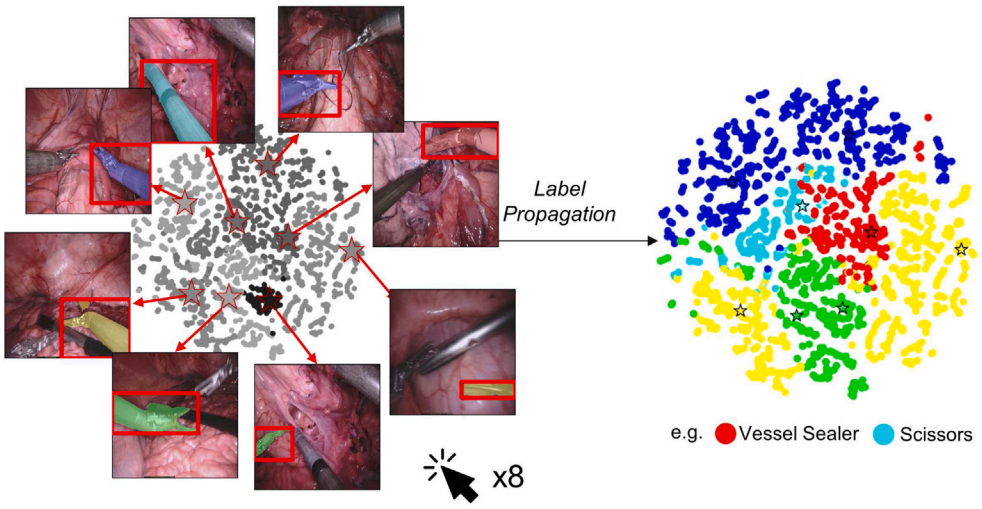

Fig. 6. Left: visualization of learnt feature representations of the EndoVis 2017 (Allan et al., 2019) training set instances, clustered (N_{km} equal to 8) and projected in the 2D space using t-SNE algorithm (Van der Maaten and Hinton, 2008). Each instance point is coloured with a different shade of grey to represent the cluster id. Prototype instance features are marked with ⋆, and the corresponding masks are overlaid on the frame and highlighted by a bounding-box, to facilitate their labelling by a human operator. The colour of the mask overlays represents the ground truth tool type that the user would assign. In practice, this annotation step can be carried out in *8 mouse clicks* only. Right: propagation of the prototype label of each centroid instance (⋆) to the other instances belonging to the same cluster. Each instance point is coloured according to its tool-type label (e.g. red: vessel sealer, light blue: scissors, etc.).

- positive examples $\{F_i^+\}$ are sampled from the instance tube T_i, built from the frame-by-frame tracking of the instance i. Such tracking is described in Fig. 5. Given the consecutive images I_t and I_{t+1}, and their corresponding sets of instrument instances, tracking is solved by projecting the centroids of I_t into I_{t+1} space using the optical flow OF, computed between I_t and I_{t+1}. Each I_t centroid is then matched to the closest I_{t+1} centroid. Optical flow projection allows us to robustly handle tool movements between consecutive frames, reducing the chances of wrong matching;
- negative examples $\{F_i^-\}$ can be sampled either from different tubes belonging to the same frame or from tubes far apart in time.

The feature extractor network is then trained by optimizing the loss L_F between $\{F_i^+\}$ and $\{F_i^-\}$:

$$L_F = L_{SCL}(\{F_i^+\}, \{F_i^-\}), \tag{3}$$

where L_{SCL} is the Supervised Contrastive Loss formulation proposed in Khosla et al. (2020), with each instance tube treated as a separate class.

Inference: once trained, given an image I, and the corresponding set of instance masks $\{\tilde{M}_i^{Inst}\}$, the feature extractor model predicts, for each instance, a feature vector F_i, encoding its semantic content.

3.3. Instance classification

The aim of this step is learning to classify each tool instance, represented by a compact feature vector, according to its tool type.

Training: let us consider the set of available tool-type classes $\{S_i\}$, with i in $[1, N_{cls}]$, with N_{cls} being the total number of tool classes. A classifier model must now be trained to learn the mapping between instance features and class labels from that set. In the absence of pixel-level semantic labels, we solve this task by only relying on tool presence labels. As tool presence labels are not spatially localized, the matching between training tool instances and tool presence labels must be defined. The class activation approach, commonly adopted for weakly-supervised object detection, requires *frame-wise* ground truth annotations about tool presence, which makes it not directly applicable to *sequence-wise* labels. We therefore propose a more flexible solution, applicable to both *frame-wise* and *sequence-wise* labels. Our solution is designed to solve the matching problem by injecting a minimal amount of human knowledge, specifically collected to maximize its information content while minimizing the annotation effort. Specifically, we automatically select a tiny number of highly representative instances (*prototype* instances) and ask a human operator to label them.

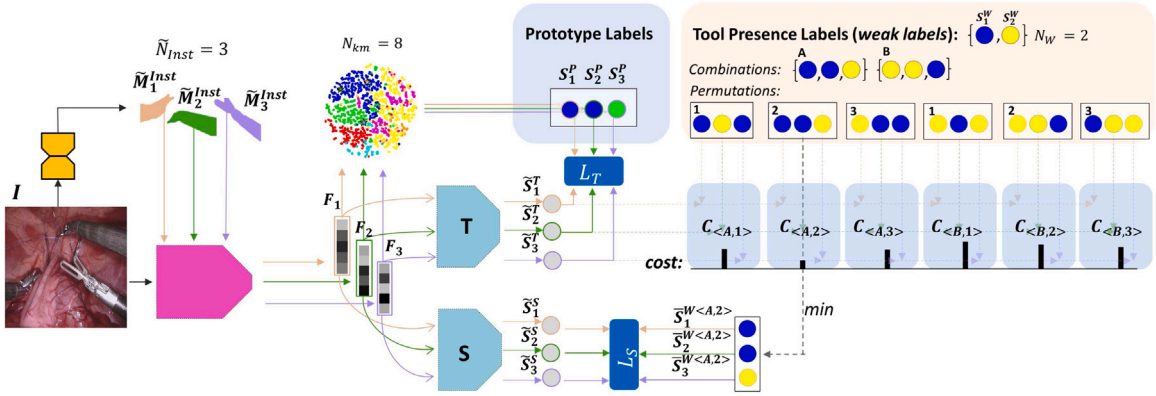

Fig. 7. Overview of the proposed weakly-supervised instance classification module. Given an image I, the corresponding set of instance-wise features $\{F_i\}$, with i in $[1, \tilde{N}_{Inst}]$, is obtained from the instance masks $\tilde{M}_i^{Inst}$. Each feature is mapped to the corresponding prototype label S_i^P, which, as shown in this case, does not necessarily correspond to the ground truth label. Each feature is also independently passed through the Teacher (**T**) and Student networks (**S**), yielding the predicted probabilities $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{T}$, $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{S}$, and the corresponding predicted labels $\tilde{S}_i^T$, $\tilde{S}_i^S$ (for the sake of readability only the latter are shown in the picture). **T** is trained optimizing the loss L_T computed using the prototype labels $\{S_i^P\}$. Simultaneously, **T** predictions are used to compute the assignment costs $C_{\langle i_C, i_P \rangle}$ for each i_P permutation of each i_C combination of the weak labels $\{S_i^W\}$, with i in $[1, N_W]$. The ordered set $[\tilde{S}_i^W]$, with i in $[1, \tilde{N}_{Inst}]$, corresponding to the minimum assignment cost, is used to compute the loss L_S for Student network optimization.

The gathered information is then used to match tool presence labels and instances, providing an effective supervision signal for classifier training. The two steps are now detailed.

(1) Prototype labelling: given the complete set of learnt features for all the instances in the training set, unsupervised clustering is applied. In our experiments, we make use of the standard K-Means++ clustering algorithm (Arthur and Vassilvitskii, 2006), with the number of clusters N_{km} regarded as a hyper-parameter. The N_{km} instances corresponding to the clusters' centroids are defined as *prototype instances*. A human operator would now be required to assign a label S^P from the set $\{S_i\}$ to each prototype instance. In order to propagate the prototype instance labels to the rest of the training instances, we require all instances belonging to the same cluster to share the same tool-type label S^P. Fig. 6 provides a visualization the of prototype instance labelling process, and of the result of prototype label propagation. In principle, a number of clusters N_{km} equal to N_{cls}, the total number of tool-type classes available, is sufficient to correctly label the whole training dataset, and potentially to directly deploy the instance segmentation model: given an unseen image I and a predicted tool instance mask $\tilde{M}_i^{Inst}$ from that image, inference would then be performed by extracting the corresponding feature vector F_i and associating it to the prototype label S^P of the cluster closest to F_i in the feature space. However, in practice, as the feature learning step is imperfect, the prototype labels can be noisy, as experimentally shown in Section 5. Nonetheless, we show that the information provided by prototype labels can be combined with tool presence labels, providing an effective supervision signal for classifier training.

(2) tool presence labels incorporation: let us define the tool presence labels associated to a certain frame as $\{S_i^W\}$, with i in $[1, N_W]$, and N_W being the total number of unique tool-type labels associated to the frame. $\{S_i^W\}$ is a subset of the set of tool-type classes $\{S_i\}$, with i in $[1, N_{cls}]$. As discussed in Section 1, tool presence labels can be defined as *frame-wise*, if they indicate which tool types are *effectively* visible in the frame, or *sequence-wise*, if they indicate which tool types are visible at some point in the sequence the frame belongs to, but not necessarily in such frame. Tool presence labels, either *frame-wise* or *sequence-wise*, do not provide tool localization information, and are therefore defined as *weak labels* for the segmentation task. While cheaply obtainable, such weak labels are often overlooked by segmentation approaches, as they pose several challenges:

- differently from pixel-level labels, tool presence labels are not directly matched to a specific instance, making them hard to digest for standard segmentation architectures, designed to learn from pixel-level annotations;
- depending on the system/annotation protocol used to collect the information, the presence of multiple instances of the same tool type may not be recorded. In the Cholec80 dataset (Twinanda et al., 2016), for example, *frame-wise* tool presence labels do not keep track of multiple tool instances;
- *sequence-wise* labels commonly do not reflect which tool types are effectively visible in each frame. In the case of robotic surgery, for example, tools are attached beforehand to the robotic system, potentially remaining unused for relatively long periods of time. Similarly for surgical phases, certain tools, like the ones used for coagulation, may be linked to every phase of a procedure, while being visible only for small amounts of time.

In order to make effective use of such information, each tool instance in a frame must be matched to a weak label from the set $\{S_i^W\}$ associated with that frame. Once the matching is found, a classifier model can be trained on the matched labels. In practice, the tool presence labels softly constrain the training of the classifier, providing a reduced set of tool-type labels among which the ground truth one for each instance is to be found.

Let us consider an image I, the sets of instance masks, features and prototype labels $\{\tilde{M}_i^{Inst}\}, \{F_i\}, \{S_i^P\}$, with i in $[1, \tilde{N}_{Inst}]$, and the set of weak labels $\{S_i^W\}$, with i in $[1, N_W]$ associated to I. Mining such weak labels requires finding the function ξ, matching the set of $\tilde{N}_{Inst}$ features to the set of N_W weak labels. However, in the most general case, such transformation is:

- *non-injective*, as there could be multiple instances sharing the same tool label S_i^W;
- *non-surjective*, as a certain tool label S_i^W may not be present in a specific frame.

This implies that given the set of $\tilde{N}_{Inst}$ predicted tool instances in a frame, different combinations of $\tilde{N}_{Inst}$ elements of the N_W weak labels are plausible. In principle, all the combinations with repetition of $\tilde{N}_{Inst}$ elements of the N_W labels are eligible, as multiple instances of the same tool type could be simultaneously present in a frame. Among the set of plausible weak label combinations, the correct label combination must be identified, and the matching between each instance and each weak label in such combination must be determined. This could be achieved by associating to each i_P permutation of each i_C plausible combination of the weak labels an assignment cost $C_{\langle i_C, i_P \rangle}$. Each couple $\langle i_C, i_P \rangle$ yields an ordered set of weak labels $[S_i^{W\langle i_P, i_C \rangle}]$, with i in $[1, \tilde{N}_{Inst}]$, where $S_1^{W\langle i_P, i_C \rangle}$ corresponds to F_1, $S_2^{W\langle i_P, i_C \rangle}$ corresponds to F_2, etc. Among them, the ordered set minimizing the assignment cost could be

selected and used for the classifier training.

To solve this problem we propose a Teacher-Student approach (Fig. 7), exploiting the knowledge gathered from the prototype labels. Teacher and Student are two identical classifiers that map a feature vector F_i to the vectors $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{T}$, $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{S}$, respectively. $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{T}$, $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{S}$ represent the predicted probability of the instance to belong to each of the N_{cls} classes, according to Teacher and Student, respectively. From $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{T}$, $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{S}$ the class with the highest probability $\tilde{S}_i^T$, $\tilde{S}_i^S$ is regarded as the predicted label. The Teacher network is trained to map each feature F_i to the corresponding prototype label S_i^P, by optimizing the instance classification loss L_{T_i}:

$$L_{T_i} = L_{CE}(\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{T}, S_i^P). \tag{4}$$

For each couple $\langle i_C, i_P \rangle$, its assignment cost $C_{\langle i_C, i_P \rangle}$ can then be computed as the average cross-entropy loss between the predicted probabilities $[\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{T}]$ and the weak labels $[S_i^{W\langle i_P, i_C \rangle}]$, corresponding to that couple, as follows:

$$C_{\langle i_C, i_P \rangle} = \frac{1}{\tilde{N}_{Inst}} \sum_{i=1}^{\tilde{N}_{Inst}} L_{CE}(\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{T}, S_i^{W\langle i_C, i_P \rangle}). \tag{5}$$

The ordered set of weak labels $[\bar{S}_i^{W, i_P, i_C}]$, corresponding to the couple $\langle i_C, i_P \rangle$ minimizing the assignment cost, is selected. The Student network is then trained by optimizing the instance classification loss L_{S_i}, between the predicted probabilities $[\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{S}]$ and the matched weak labels $[\bar{S}_i^{W, i_P, i_C}]$:

$$L_{S_i} = L_{CE}(\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{S}, \bar{S}_i^{W\langle i_C, i_P \rangle}). \tag{6}$$

In practice, the Teacher network applies the knowledge gathered from the prototype labels to identify the correct ordered set of weak labels used for Student training. Doing so, the Teacher approximates the function ξ, matching each of the $\tilde{N}_{Inst}$ tool instances to a weak label from the set $\{S_i^W\}$.

This general framework applies to both *frame-wise* and *sequence-wise* tool presence labels. In the case of *frame-wise* labels, ξ becomes surjective, significantly reducing the space of possible solutions and facilitating the matching.

Inference: once trained, only the Student classifier is required to perform inference. Specifically, given a tool instance encoded in the feature vector F_i, the trained Student classifier predicts $\tilde{\mathbf{P}}_\mathbf{i}^\mathbf{S}$, representing the probability of the instance to belong to each of the N_{cls} tool-type classes. The complete inference pipeline can be seen in Fig. 2, bottom.

4. Experimental set-up

The proposed framework was validated on the MICCAI 2017 and 2018 EndoVis Robotic Instrument Segmentation Challenge datasets. The two datasets are now introduced (Section 4.1), together with the specific design choices and training details (Section 4.2).

4.1. Datasets

EndoVis2017 (Allan et al., 2019): the original challenge dataset consists of 10 video clips, resampled at a frame rate of 1 frame-per-second, of abdominal porcine procedures, performed using the da Vinci robotic system. Each clip contains 300 high-resolution frames (1024 × 1280). During the challenge 8 × 225 frames were released for training, while the remaining 8 × 75 frames and two additional clips were held out by the organizers for testing. A total of 7 tool classes are present in the dataset. We provide results on this dataset according to the same evaluation protocol as Shvets et al. (2018), by performing 4-fold cross-validation on the 8 × 225 released training data (regrouped in 4 splits). We report the average metric over the 4 splits, for direct comparison with state-of-the-art approaches.

EndoVis2018 (Allan et al., 2020): the original challenge dataset contains 19 video clips, resampled at a frame rate of 1 frame-per-second, of abdominal porcine procedures, performed using da Vinci robotic system. Each video contains a total of 300 high-resolution frames (1024 × 1280). During the challenge, 15 clips were released for training, while the remaining clips were held out by the organizers for testing. The dataset was originally annotated for anatomy and tool-part segmentation and did not feature tool-type labels. González et al. (2020) annotated with pixel-level semantic labels 149 frames for each of the 15 training clips, and split them into a training set consisting of 11 clips, and a validation set containing the remaining 4 clips. The same 7 tool classes from EndoVis2017 dataset were used. We provide results on this dataset according to the same evaluation protocol as González et al. (2020), by training on the 11 training clips, and validating on the remaining 4 clips.

As the proposed SAF-IS approach requires binary instrument masks to train, we provide results using both manually annotated binary masks and automatically segmented masks generated using the unsupervised FUN-SIS approach (Sestini et al., 2023). The mean binary IoU for the FUN-SIS approach on the EndoVis2017 and EndoVis2018 datasets is equal to 83.7% and 81.3%, respectively.

Frame-wise tool presence labels were automatically generated for each frame as the unique pixel-level semantic labels present in the corresponding ground truth semantic masks. *Sequence-wise* tool presence labels were also automatically generated, by considering each video clip in the datasets as a sequence, and assigning to each clip, as *sequence-wise* labels, the full set of unique semantic labels present in the ground truth semantic masks of all the frames in the clip. For 46.12% of the frames in the EndoVis2018 dataset the *sequence-wise* labels do not correspond to the *frame-wise* labels (40.72% for EndoVis2017 dataset), i.e., for a certain frame, its *sequence-wise* labels contain at least a tool type which is not visible in it (but which is present at some point in the clip it belongs to).

4.2. Design choices & training details

Tool instantiation: the instantiation model is implemented as a U-Net architecture with SegFormer encoder (Xie et al., 2021), available from the *Segmentation Models* library in PyTorch. The training was carried out for 60 epochs using the Adam optimizer with a learning rate equal to 1e−3 and a batch size of 32, applying standard photometric and geometric augmentations from the *Albumentation* library to the original images, resized to a 256 × 256 resolution. During inference, centroids were selected by overlapping the predicted displacement field with a square grid of 32 × 32 resolution (i.e. each grid square of 8 × 8 pixel dimension); a threshold ε_C of 5 was used to select centroid squares (i.e. squares with a per-pixel average of at least 5 displacement vectors pointing at them were selected as centroids). The impact of grid resolution and threshold value is investigated in Section 6.

Feature representation learning: the feature extractor network is implemented as a ResNet-50 architecture. Each instance mask is multiplied by the output of the $conv3\_4$ layer. Instance-wise features are obtained by applying a global average pooling to the output of the $conv5\_3$ layer, having 2048 feature channels. The training was carried out for 80 epochs using the Adam optimizer with a learning rate equal to 5e−5 and a batch size of 64, applying standard photometric and geometric augmentations to the original images, resized to a 512 × 512 resolution. For the contrastive loss L_{SCL} a temperature factor equal to 0.1 was used. Optical flow computation, required for positive instance sampling, was carried out using RAFT (Teed and Deng, 2020), a state-of-the-art model trained on the publicly available non-surgical dataset FlyingThings (Mayer et al., 2016).

Instance classification: for the main experiments (Section 5.2), K-Means++ clustering algorithm was applied with a total number of clusters N_{km} equal to 8 (therefore 8 instances were required to be labelled by a human user). While in the real scenario such assignment

Table 1
Tool instantiation results for the proposed SAF-IS approach and Mask-RCNN on EndoVis 2017 and 2018 datasets, trained according to three modalities: fully-supervised (GT) and unsupervised using Connected Component labelling of manually annotated masks (CC_M) and FUN-SIS predicted masks (CC_F).

Superv.	Method	EndoVis			
		2017		2018	
		AP@0.5	AP@0.7	AP@0.5	AP@0.7
GT	MRCNN	76.11	61.87	75.01	63.12
	SAF-IS	**88.40**	**72.12**	**78.57**	**66.00**
CC_M	MRCNN	71.26	55.98	73.99	60.04
	SAF-IS	**85.36**	**63.70**	**75.92**	**61.08**
CC_F	MRCNN	63.81	44.99	62.48	42.31
	SAF-IS	**81.31**	**56.14**	**71.01**	**49.17**

would be performed by a human operator, as discussed in Section 6, it was here automatically performed by associating to each prototype instance the tool-type label of the ground truth instance of the same frame having the maximum overlap according to the Intersection-over-Union metric. An extended explanation of the label matching problem, as well as the implementation details, are provided in Appendix A.

The classification networks (Teacher, Student) were implemented as a 2-layer fully-connected network, with an intermediate feature size of 512 and batch normalization. The training was carried out for 40 epochs using the Adam optimizer with a learning rate equal to 1e−4 and a batch size of 128, applying standard photometric and geometric augmentations to the original images, resized to a 512×512 resolution.

5. Experiments and results analysis

We now present the experimental validation of the proposed SAF-IS framework and compare it with state-of-the-art approaches. Tool instantiation results and complete instance segmentation results are separately presented in Sections 5.1 & 5.2, respectively.

5.1. Tool instantiation

In order to analyse tool instantiation quality, we evaluate results according to a class-agnostic Average-Precision metric, computed for two values of threshold Intersection-Over-Union (IoU): AP@0.5 (50%), AP@0.7 (70%). We present results obtained by our unsupervised approach using, as binary masks, both manual annotations (SAF-IS CC_M) and unsupervised FUN-SIS predictions (SAF-IS CC_F). In addition, we report results for the instantiation model trained in a fully-supervised manner on the ground truth displacement field (SAF-IS GT). To the best of our knowledge, no other work has previously tackled the problem of learning to extract individual tool instance masks from an image with no additional supervision other than the binary segmentation mask. For this reason, we compare our solution against a Mask-RCNN baseline, trained under the same fully-supervised (MRCNN GT) and unsupervised modalities (MRCNN CC_M, MRCNN CC_F). However, as Mask-RCNN is an anchor-based approach, the local masking for automatically identified overlapping tools (M^{OV}, described in Section 3.1), is not easily implementable and would require substantial architectural modifications which are beyond the scope of this work. Therefore we limit the augmentation strategy for unsupervised Mask-RCNN experiments to instance pasting, described in Section 3.1.

Results presented in Table 1 show how our proposed solution outperforms Mask-RCNN across both datasets and for all three training modalities. A similar result for the fully-supervised training modality was already presented in Kurmann et al. (2021). These experiments highlight the benefits of tool instantiation based on direct centroid regression, beyond full supervision, for the unsupervised setting. Indeed, the unsupervised SAF-IS solution using binary annotated masks (SAF-IS CC_M) closely follows the fully-supervised one (SAF-IS GT), with an average gap of $-\Delta 3.3\%$ AP@0.5 across the two datasets. In addition, the greatest performance gap between SAF-IS and Mask-RCNN is found when using FUN-SIS binary masks to train (CC_F): $+\Delta 17.5\%$ AP@0.5 and $+\Delta 11.15\%$ AP@0.7, in the EndoVis 2017 dataset. This result shows how our solution is particularly suitable to handle a noisy supervision signal. Finally, the performance gap between SAF-IS CC_F and SAF-IS CC_M is significantly smaller for the AP@0.5 metric ($-\Delta 4.48\%$ on average across the two datasets) compared to the AP@0.7 metric ($-\Delta 9.74\%$). This can be attributed to the lower quality of FUN-SIS binary segmentation masks, causing a performance drop when a high IoU threshold is used: the lower 50% IoU threshold, instead, being less affected by possible inaccuracies in the binary segmentation masks, highlights the high instantiation quality.

5.2. Tool instance segmentation

In order to evaluate instance segmentation results, and compare them with other state-of-the-art segmentation approaches, we adopt the commonly used IoU EndoVis challenge metric defined in González et al. (2020). It is worth noticing that such metric treats the segmentation problem as pixel-wise classification, without providing information about instantiation quality.

Table 2 reports the results of our SAF-IS framework and for several state-of-the-art solutions. For each method, the table highlights the type of supervision used for training. State-of-the-art approaches are all trained in a fully-supervised manner using pixel-level semantic annotations (S), in combination with pixel-level instance annotations for instance segmentation methods (I). Our SAF-IS framework does not require pixel-level semantic or instance annotations to train, relying instead only on prototype instance labels (P) - 8 for the experiments reported in this Table - and weak labels, in the form of *frame-wise* (FW) or *sequence-wise* (SW) tool presence labels (results for both modalities are reported). In addition, SAF-IS can be trained using manually annotated binary masks (B) if available, or rely on the predictions of the unsupervised FUN-SIS approach (results for both modalities are also reported).

Results presented in Table 2 show that our SAF-IS approach, trained using only tool presence labels and 8 prototype labels, outperforms fully-supervised and semi-supervised solutions adopting a semantic segmentation problem formulation (Ternaus, MF-TN, DMF-TN), despite not requiring any spatial annotation. On the EndoVis 2017 dataset our solution also outperforms a standard Mask-RCNN (MRCNN), trained on manually annotated segmentation masks and bounding-boxes for ground truth instantiation. In addition to pixel-level semantic and instance annotations, the solutions outperforming our SAF-IS approach also rely on temporal information during inference (†) and additional tool-part segmentation annotations (‡). It is worth noticing that temporal modelling is a natural extension for SAF-IS, as tool tracking information is already extracted as part of the instance-wise feature learning step. Finally, a comparison between SAF-IS models trained on *frame-wise* (FW) and *sequence-wise* (SW) tool presence labels, shows the effectiveness of our Teacher-Student solution to extract a reliable supervision signal from the automatically obtainable *sequence-wise* labels, with an average gap between the two of less than 1.2% IoU, across datasets and binary mask sources. Qualitative results are shown in Figs. 12 & 13 at the end of the manuscript.

The full inference pipeline is composed by three sequential models (tool instantiation model, feature extractor, classifier). Given a frame resolution of 512×512, the average per-frame inference time on a single NVIDIA V100 GPU for the three models is 85 ms for the instantiation model, 13.15 ms for the feature extractor model, 0.6 ms for the classification model (considering an average of 2.43 instances simultaneously classified). The total inference time for one frame is therefore 98.75 ms (~10.13 fps) on a single NVIDIA V100 GPU, without any specific optimization. For comparison Ternaus (Shvets et al., 2018), M&C (Kurmann et al., 2021), Tra-SeTr (Zhao et al., 2022) report an

Table 2
Instance segmentation results for the proposed SAF-IS approach, state-of-the-art methods on EndoVis 2017 and 2018 datasets, evaluated according to the IoU EndoVis challenge metric defined in González et al. (2020). Supervision signals used by each approach are reported: pixel-level semantic labels (S, percentage of labelled data reported for semi-supervised approaches), pixel-level instance labels (I), required by fully-supervised instance segmentation approaches, binary segmentation masks (B, for SAF-IS, if not checked FUN-SIS predicted masks are used), prototype labels (P, 8 labels in total in these experiments, ∼0.3% of total training instances), frame-wise tool presence labels (FW) and sequence-wise tool presence labels (SW).

Method	Supervision type						EndoVis	
	Pixel-level			Weak			2017	2018
	S	I	B	P	FW	SW		
Ternaus (Shvets et al., 2018)	✓						33.78	/
MF-TN[a] (Jin et al., 2019)	✓						36.62	/
DMF-TN[a] (Zhao et al., 2020)	$✓_{30\%}$						**45.83**	/
DMF-TN[a] (Zhao et al., 2020)	$✓_{20\%}$						43.71	/
DMF-TN[a] (Zhao et al., 2020)	$✓_{10\%}$						33.64	/
M&C[ab](Kurmann et al., 2021)	✓	✓					**65.70**	/
ISI-Net[a] (González et al., 2020)	✓	✓					55.62	73.03
MRCNN (Kong et al., 2021)	✓	✓					42.28	/
Tra-SeTr[a] (Zhao et al., 2022)	✓	✓					60.04	**76.20**
SAF-IS			✓	$✓_{0.3\%}$			43.86	56.62
SAF-IS			✓	$✓_{0.3\%}$	✓		**53.73**	63.38
SAF-IS			✓	$✓_{0.3\%}$		✓	52.64	**63.57**
SAF-IS				$✓_{0.3\%}$			30.47	54.08
SAF-IS				$✓_{0.3\%}$	✓		**45.86**	**58.03**
SAF-IS				$✓_{0.3\%}$		✓	42.41	57.75

[a] Methods using temporal information at inference time.
[b] Methods using additional tool-part annotations for training.

Table 3
Results of the ablation study on unsupervised instrument instantiation from manually annotated (a) and FUN-SIS predicted (b) binary masks, highlighting the separate and combined impact of masking potentially overlapping instances (OV) and pasting random tool instances (PS).

(a)

Augm.		EndoVis			
OV	PS	2017		2018	
		AP@0.5	AP@0.7	AP@0.5	AP@0.7
		74.85	56.585	71.56	58.08
✓		77.74	54.82	**77.58**	62.00
	✓	81.91	59.82	70.54	57.98
✓	✓	**85.35**	**63.70**	75.92	**62.08**

(b)

Augm.		EndoVis			
OV	PS	2017		2018	
		AP@0.5	AP@0.7	AP@0.5	AP@0.7
		67.82	47.86	65.23	43.94
✓		71.80	45.69	**71.91**	48.99
	✓	72.41	49.42	67.99	47.12
✓	✓	**81.31**	**56.14**	71.01	**49.16**

inference speed of 5.78 fps (NVIDIA 1080Ti GPU), 15 fps (NVIDIA 2080 Ti GPU) and 23 fps (a NVIDIA Titan Xp GPU), respectively. While such time may not be suitable for applications that require near-zero delay (e.g. augmented reality), it is compatible with online applications that do not require zero-delay feedback. Finally, the total training time for the tool instantiation model, feature extractor, Teacher and Student classifiers, on the EndoVis2017 dataset, is ∼16 h on a single NVIDIA V100 GPU.

6. Ablation studies

In order to provide a deeper insight into the SAF-IS framework, we now present and discuss ablation studies on three critical design choices: the augmentation strategy for tool instantiation, the inference parameters for tool instantiation and the number of prototype labels required for instance classification.

6.1. Tool instantiation augmentation strategy

In order to train the displacement network for instrument instantiation, a pseudo-supervision signal is generated from the binary masks using a Connected Component algorithm. Such signal is subsequently refined by (1) preventing training on potentially overlapping instances (OV) and (2) pasting random tool instances (PS) to artificially simulate the case of overlapping instances (Section 3.1).

Table 3 provides the results of an ablation study exploring different combinations of the two augmentation strategies. Such results show the effectiveness of the two augmentation strategies, and of their simultaneous use. In the case of binary annotated masks, instance masking (OV) provides an average improvement of $+\Delta 4.46\%$ AP@0.5 and $+\Delta 1.08\%$ AP@0.7 across the two datasets, compared to the setting where no augmentation is used; instance pasting (PS) provide an average improvement of $+\Delta 3.03\%$ AP@0.5 and $+\Delta 1.56\%$ AP@0.7; the two strategies combined provide an average improvement of $+\Delta 7.02\%$ AP@0.5 and $+\Delta 5.55\%$ AP@0.7. On the EndoVis 2018 dataset, paste augmentation appears less effective: this could be due to the fact that several frames in it present at least 4 separate tool instances, making the additional pasting redundant, and potentially detrimental as frames can become too cluttered.

6.2. Tool instantiation inference parameters

In order to obtain instance masks, a square grid is overlapped to the predicted displacement field; centroid squares are then selected as the ones whose per-pixel average of vectors pointing inside them is greater than the threshold value ε_C. The grid resolution (equal to 32×32 in our main experiments) and the threshold ε_C (equal to 5 in our main experiments) regulate the trade-off between precision and recall of the obtained instance masks. We experimentally evaluate the impact of the two parameters by varying them in a grid-like manner, with grid resolution in $[8, 16, 32, 64, 128]$ and ε_C in $[1, 3, 5, 7, 10]$. Their different combinations are used to obtain instance masks from the same displacement fields. The AP@0.5 between the obtained masks and the ground truth instances is reported in Fig. 8 for both the EndoVis2017 and EndoVis2018 datasets.

The presented results, together with the qualitative results shown in Fig. 9, clearly highlight the impact of the two parameters. For intermediate grid resolution values (32×32, 64×64), the impact of

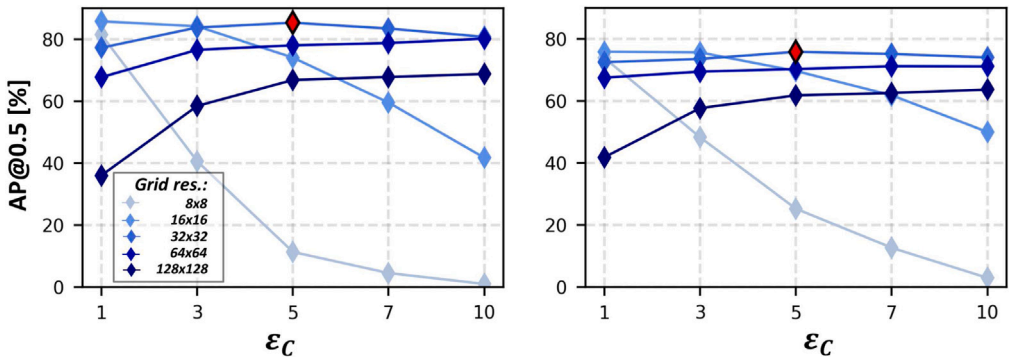

Fig. 8. Impact of grid square resolution and threshold value ε_C on the tool instantiation quality for the EndoVis2017 dataset (left) and EndoVis2018 dataset (right). The combination used in our main experiments is highlighted in red.

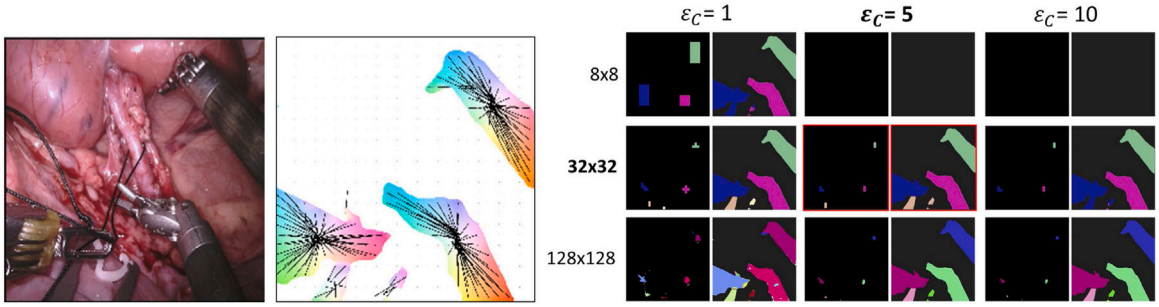

Fig. 9. From left to right, original image, predicted displacement field, and examples of centroid regions and instantiation masks for different combinations of grid resolution and threshold ε_C. Mask colours indicate the ID assigned to the tube each instance belongs to. The combination adopted in our main experiments is highlighted in red.

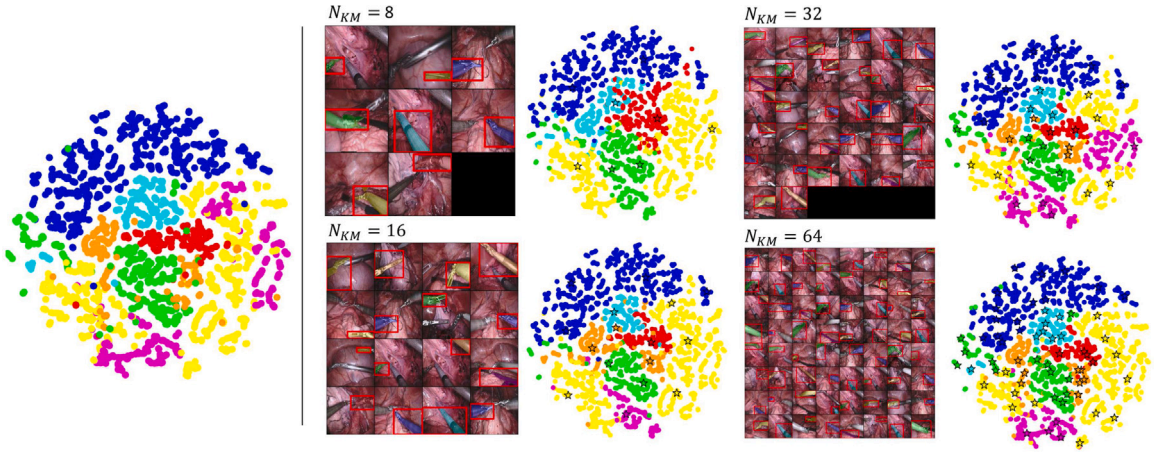

Fig. 10. Left: visualization of the learnt feature representations of the EndoVis 2017 training set instances, projected in the 2D space using t-SNE algorithm (Van der Maaten and Hinton, 2008). Each instance point is coloured according to the corresponding ground truth tool class. Right: K-Means++ clustering and prototype labels obtained using different numbers of clusters N_{km}; projected features and prototype instances are coloured accordingly to the corresponding prototype labels.

ε_C is minimal. However, as the grid solution decreases (16×16, 8×8), a high value of ε_C negatively affects the quality as instantiation, as the average convergence rate on large squares tends to be lower. This can be also observed from the qualitative instantiation results shown in Fig. 9, top-right, where no candidate squares reach the threshold. Vice-versa, high grid resolution values (128×128) tend to be more negatively affected by a low ε_C, as it leads to the identification of many false positive centroids (instantiation results from Fig. 9, bottom-left).

6.3. Prototype labels number

In SAF-IS, the Teacher network is required to gather knowledge from the prototype labels, in order to be able to identify the correct ordered sets of weak labels used for Student training. Prototype labels, therefore can have a crucial influence on the quality of instance classification. In addition, they represent the manual annotation necessarily required by SAF-IS for training, as both binary tool masks and tool presence labels can be automatically obtained. Therefore we now present, in Table 4, the impact on the segmentation performance, of the number of clusters N_{km} used for K-Means clustering, equal to the number of prototype labels assigned by a human operator. In order to provide a complete overview, we present segmentation results obtained via instance classification by direct K-Means inference, Teacher classifier prediction and Student classifier prediction, when trained using *sequence-wise* or *frame-wise* tool presence labels. In addition, Fig. 10 provides a visualization of the learnt feature distribution, the clustering

Table 4
Instance segmentation results of the ablation study investigating the impact of the number of clusters N_{km} on final segmentation results, using, for instance classification, direct K-Means inference, Teacher predictions and Student predictions, trained using *sequence-wise* (SW) or *frame-wise* (FW) tool presence labels. Results obtained using (a): manually annotated binary masks on the EndoVis2017 dataset, (b): FUN-SIS predicted binary masks on the EndoVis2017 dataset, (c): manually annotated binary masks on the EndoVis2018 dataset, (d): FUN-SIS predicted binary masks on the EndoVis2018 dataset. Segmentation results were evaluated using the challenge IoU metric. The best results across the number of clusters are highlighted in bold.

(a)

N_{km}	K-Means	Teacher	Student	
			SW	FW
8	43.86	45.66	52.64	**53.73**
16	38.52	41.34	50.02	52.64
32	42.33	45.26	51.23	52.44
64	**44.76**	**48.38**	**52.84**	53.37

(b)

N_{km}	K-Means	Teacher	Student	
			SW	FW
8	30.47	30.88	42.21	45.86
16	**37.86**	**41.81**	**46.95**	47.33
32	32.49	36.40	46.40	**48.00**
64	36.10	41.02	46.91	47.96

(c)

N_{km}	K-Means	Teacher	Student	
			SW	FW
8	**56.62**	56.80	**63.57**	63.38
16	56.03	57.25	60.63	61.96
32	56.11	**57.48**	62.80	**64.76**
64	53.80	57.22	62.24	63.88

(d)

N_{km}	K-Means	Teacher	Student	
			SW	FW
8	54.08	54.14	57.75	58.03
16	**56.53**	**57.04**	57.40	58.53
32	55.53	55.86	**58.45**	59.48
64	55.52	56.02	57.92	**59.85**

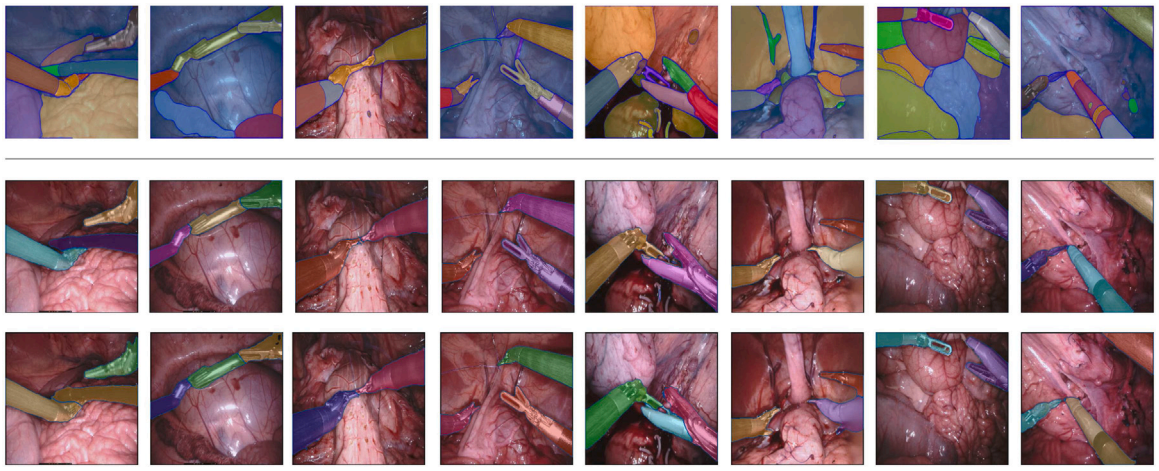

Fig. 11. Top row: SAM (Kirillov et al., 2023) segmentation results on the EndoVis2017 dataset. Central row: SAF-IS instantiation results obtained from binary manually annotated masks. Bottom row: SAF-IS instantiation results obtained from FUN-SIS predicted binary masks. Each instance is coloured using a random colour, which is not meant to represent tool-type classes.

process and the automatically selected prototype instances.

Result analysis provides different insights into the method. First of all, although a marginal improvement exists, increasing the number of prototype instances does not provide substantial performance gains for the Student network. This result may indicate that effective feature learning is a crucial methodological bottleneck, that cannot be solved by simply increasing the number of human-assigned labels. Secondly, the presented results highlight the consistent improvement in performance provided by the Student network, trained on the weak labels matched through the Teacher model. Although the Teacher learns to substantially replicate K-Means clustering classification, as shown by their similar performance, this is enough to perform a good weak label matching, responsible for the Student's superior performance.

7. Discussion

The results presented in Sections 5 & 6 confirm the soundness of the proposed SAF-IS framework for instance segmentation. Our solution trains on endoscopic videos paired with binary segmentation masks, potentially obtained in an unsupervised way, and is designed to incorporate tool presence labels, either *frame-wise* or *sequence-wise*. Human annotation effort can here be limited to labelling a tiny set of prototype instances, automatically selected by our approach, with inexpensive classification labels: the ablation study presented in Section 6 shows that the size of such set can be reduced to 8 instances (~0.26% of the total number of training instances), with no significant performance drop. This result goes significantly beyond existing semi-supervised solutions like (Zhao et al., 2020), where a significant set of frames (up to 30%) needs to be labelled with pixel-level annotations, while still providing inferior segmentation performance. Indeed, our complete spatial annotation-free solution, using FUN-SIS predicted binary masks for training, outperforms fully-supervised and semi-supervised semantic segmentation approaches like MF-TN and DMF-TN by a consistent margin on the EndoVis 2017 dataset. Furthermore, our SAF-IS framework effectively incorporates *sequence-wise* tool presence labels, commonly overlooked in the literature. This small gap in performance between *frame-wise* and *sequence-wise* training modalities (Table 2), shows that *sequence-wise* labels can be an effective source of supervision, while being completely free to collect.

Although a performance gap still exists with top-performing fully-supervised instance segmentation approaches, we believe there exist several directions of improvement to close such a gap. First of all, temporal modelling could be easily learnt from the already available tracking information, currently exploited only at training time for feature learning. Secondly, as highlighted by the ablation study on cluster number, feature learning represents a crucial methodological

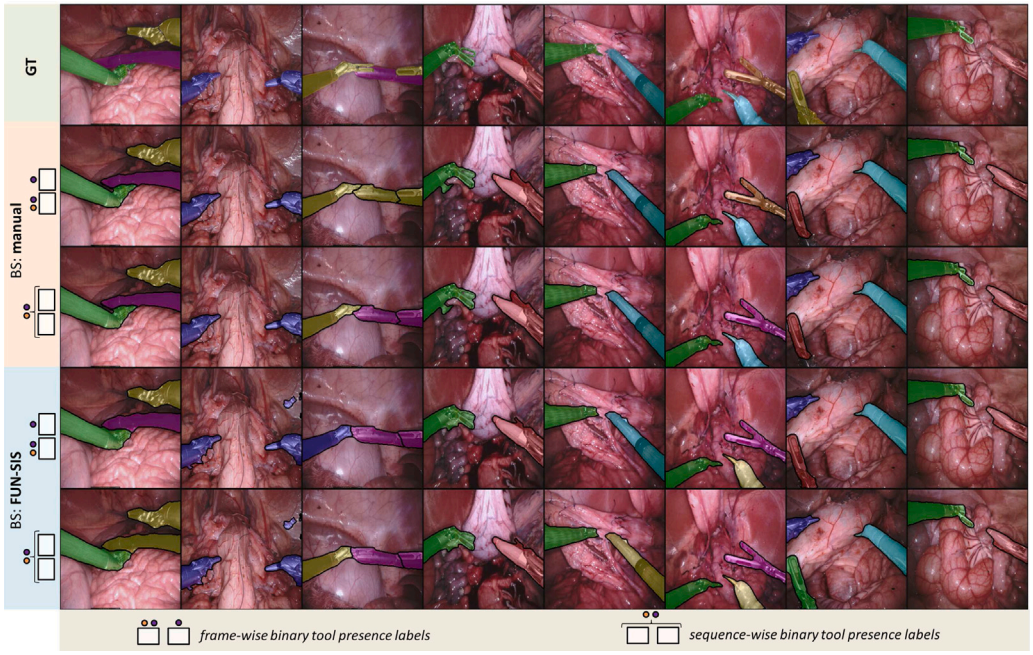

Fig. 12. Qualitative segmentation results from the EndoVis2017 dataset, highlighting, for our SAF-IS approach, the source of binary segmentation masks (BS) and the type of tool presence labels (frame-wise or sequence-wise). All the SAF-IS results are obtained using 8 prototype labels. Row 1: ground truth; rows 2-5: SAF-IS Student trained on (2) manually annotated binary masks and *frame-wise* tool presence labels, (3) manually annotated binary masks and *sequence-wise* tool presence labels, (4) FUN-SIS predicted binary masks and *frame-wise* tool presence labels, (3) FUN-SIS predicted binary masks and *sequence-wise* tool presence labels.

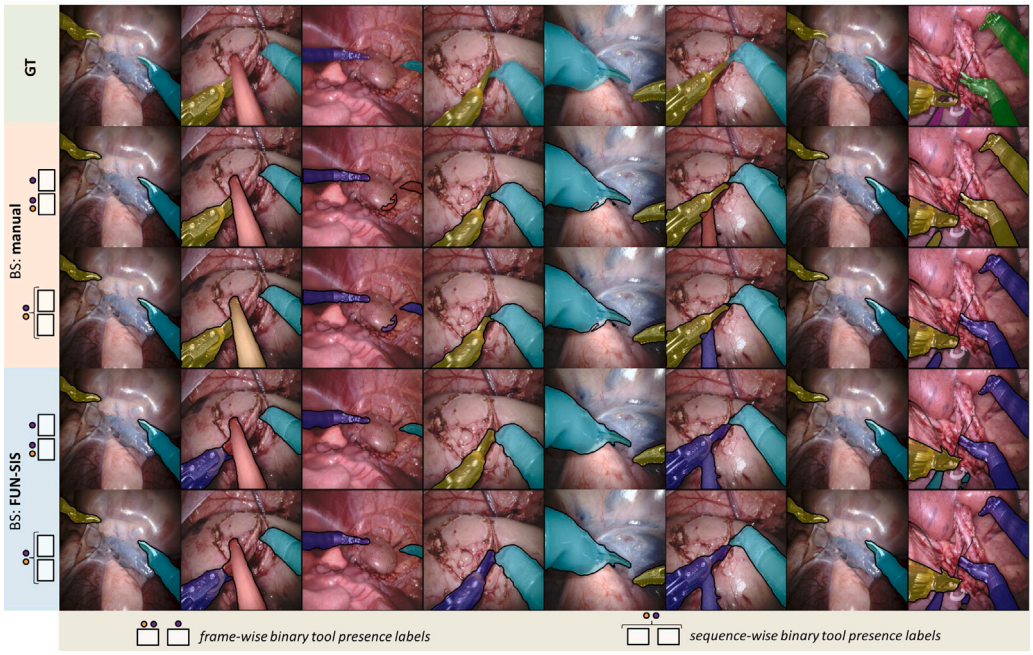

Fig. 13. Qualitative segmentation results from the EndoVis2018 dataset. Ground truth and SAF-IS Student results presented in the same order as Fig. 12 above.

bottleneck: if the learnt feature representations are sub-optimal, the unsupervised clustering may fail to separate tools belonging to different classes, hindering the following classifier training. In the current implementation, feature learning is performed in a completely unsupervised way, with no help from external information. Weak information about tool presence may be included at this stage to perform a more informed positive and negative feature sampling.

Beyond the technical improvements listed above, future work should be also directed towards extending validation to different datasets and, potentially, to different model architectures. In fact, it is worth highlighting that SAF-IS is a framework for instance segmentation model training, and as such, can work with different model architectures, beyond the ones selected for our experiments (e.g. ResNet-50 as a feature extractor). For example, lighter architectures could be selected for increased inference speed if needed. Inference speed could also be improved by using optimization techniques such as model pruning and quantization (Liang et al., 2021), and hardware optimized GPU libraries.

In future work, SAF-IS could also be adapted to perform bounding-box detection, and potentially benchmarked against weakly-supervised approaches for tool detection, like (Nwoye et al., 2019), using a similar problem formulation.

Finally, SAF-IS, not requiring pixel-level labels, can leverage recent breakthrough solutions like SAM (Segment Anything Model, Kirillov et al. (2023)). Fig. 11 shows qualitative results from SAM automatic segmentation on the EndoVis2017 dataset, compared to SAF-IS predictions. Even if SAM automatic segmentation results are currently over-segmenting tools, breaking them up into individual parts, our SAF-IS instantiation predictions could be used to group these parts, exploiting the high-quality boundary segmentation that SAM can already provide. Moreover, point-prompting, requiring a minimum annotation effort, can be used with SAM for improved segmentation results. In either case (automatic segmentation or point-prompting) SAM could be directly integrated into SAF-IS, providing instance-wise masks for the following feature learning and tool classification training steps.

In conclusion, SAF-IS major contribution lies in its ability to lift the need for spatial annotation of the training data. This may open up new research directions aimed at better exploiting human annotation effort, for example by focusing it on particularly representative or challenging samples.

8. Conclusion

In this work, we developed and validated SAF-IS, a Spatial Annotation Free framework for Instance Segmentation of surgical instruments. The proposed framework embraces recent breakthrough solutions for unsupervised binary segmentation, building on top of them to perform instance segmentation without requiring pixel-level semantic or instance annotations to train. Instead, SAF-IS exploits the binary tool masks to learn to encode each instance in a compact feature representation, and solves the instance classification problem by relying on cheaply obtainable tool presence labels.

In conclusion, we hope this work can show the potential of prior knowledge and weakly-supervised training for tool instance segmentation, encouraging the search for alternatives to full supervision for increasingly complex surgical computer vision tasks.

CRediT authorship contribution statement

Luca Sestini: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Benoit Rosa:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Elena De Momi:** Writing – review & editing, Supervision. **Giancarlo Ferrigno:** Writing – review & editing, Supervision. **Nicolas Padoy:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the ATLAS project. The ATLAS project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 813782. This work was also partially supported by French State Funds managed by the Agence Nationale de la Recherche (ANR), France through the Investissements d'Avenir Program under Grant ANR-11-LABX-0004 (Labex CAMI) and Grant ANR-10-IAHU-02 (IHU-Strasbourg), by French state funds managed by the ANR, France under references ANR-20-CHIA-0029-01 (National AI Chair AI4ORSafety) and ANR-18-CE19-0012, and by BPI France under reference DOS0180017/00 (project 5G-OR).

Appendix A. Tool presence labels matching

A.1. Combination set definition

Matching tool presence labels with instances is a crucial methodological step, necessary to train the instance classifier (Section 3.3).

Given an image I, the set of encoded features $\{F_i\}$, with i in $[1, \tilde{N}_{Inst}]$, and the set of weak labels $\{S_i^W\}$, with i in $[1, N_W]$ associated to I, each instance must be associated with a weak label in order to train the classifier. In other words, the problem requires identifying the ordered set $[S_i^W]$, with i in $[1, \tilde{N}_{Inst}]$, where S_1^W corresponds to F_1, S_2^W corresponds to F_2, etc.

Given the set of $\tilde{N}_{Inst}$ tool instances, different combinations of $\tilde{N}_{Inst}$ elements of the N_W weak labels are plausible. In principle, all the combinations with repetition of $\tilde{N}_{Inst}$ elements of the N_W labels are eligible. However, in practice, in our implementation, we restrict the search space by only considering the combinations containing the highest possible number of different tool-type labels from the weak labels set. Specifically, we identify the set of plausible weak label combinations as follows:

- if $\tilde{N}_{Inst} < N_W$, all the possible combinations without repetitions of $\tilde{N}_{Inst}$ elements of the N_W labels are selected (Fig. A.1, left);
- if $\tilde{N}_{Inst} > N_W$, all the possible combinations containing all the N_W labels are selected. In each combination, the remaining $\tilde{N}_{Inst} - N_W$ labels are repetitions of the N_W labels (Fig. A.1, middle);
- if $\tilde{N}_{Inst} == N_W$, the set of N_W labels is the only selected combination (Fig. A.1, right).

Although this solution may generate wrong associations in some cases (whenever multiple instances of the same tool are present in the image, and $\tilde{N}_{Inst} \leq N_W$), it ensures that also the less represented tool-type classes get assigned and learnt by the classifier. In practice, this strategy prevents the classifier from learning to predict only the most represented tool classes, by encouraging it to predict also less represented classes.

We investigate the impact of such a strategy by performing a study on the EndoVis2018 dataset, comparing the use of the restricted set of combinations (as detailed above, and done for the main experiments reported in the manuscript) with the use of the complete set of combinations. The results of such experiments are reported in Table A.1. It can be noticed how the use of the complete set of combinations, although theoretically correct, leads to a systematically lower performance for both *frame-wise* and *sequence-wise* labels.

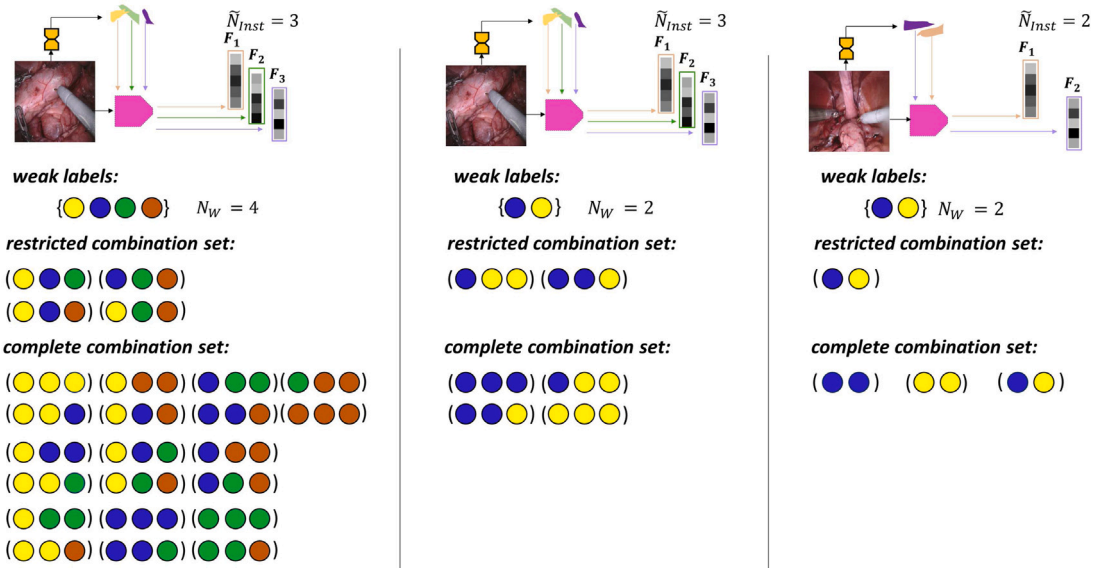

Fig. A.1. Combination set definition for tool presence label matching, for three different cases. For each case, we highlight the number of instances $\tilde{N}_{Inst}$, the weak labels set associated to the frame and the number of weak labels N_W, the complete combination set, and the restricted combination set, obtained as discussed in the text. Each colour represents a different tool type.

Table A.1
Results of the ablation study comparing the impact on the final segmentation performance of using the restricted set of combinations (as done for the main experiments reported in the manuscript, e.g. Table 2) *vs.* using the complete set of combinations for the matching problem. Results were obtained using the Student classifier, trained on the EndoVis2018 dataset using different numbers of clusters N_{km}, *sequence-wise* (SW) or *frame-wise* (FW) tool presence labels, manually annotated binary masks or FUN-SIS predicted binary masks. Segmentation results were evaluated using the challenge IoU metric. **(a)** Tool presence label matching carried out using the complete combination set approach. **(b)** Tool presence label matching carried out using the restricted combination set approach. **(c)** Relative improvement brought by using the restricted combination set approach (b) compared to using the complete combination set approach (a).

(a) Complete combination set

N_{km}	Manual		FUN-SIS	
	SW	FW	SW	FW
8	62.93	60.69	55.93	55.10
16	57.82	58.44	58.58	55.74
32	59.54	59.49	57.67	57.91
64	61.04	60.16	57.78	58.91

(b) Restricted combination set

N_{km}	Manual		FUN-SIS	
	SW	FW	SW	FW
8	63.57	63.38	57.75	58.03
16	60.63	61.96	57.40	58.53
32	62.80	64.76	58.45	59.48
64	62.24	63.88	57.92	59.85

(c) Relative difference between (b) and (a), computed as $(b_i - a_i)/a_i \cdot 100$, with a_i, b_i being corresponding entries in tables (a) and (b).

N_{km}	Manual		FUN-SIS	
	SW	FW	SW	FW
8	+1.02%	+4.43%	+3.25%	+5.32%
16	+4.86%	+6.02%	−2.01%	+5.00%
32	+5.48%	+8.86%	+1.35%	+2.71%
64	+2.62%	+6.18%	+0.59%	+1.60%

A.2. Instance classification problem interpretation

The instance classification problem could be solved using a standard Hungarian algorithm if the following two conditions were verified:

- weak labels were available at inference time;
- a cost assignment function was known.

However, none of the two assumptions holds in practice: the assignment cost is not known a-priori, and we assume that weak labels are not always available at inference time (e.g. in the case of non-robotic laparoscopy).

Therefore, as presented in Section 3.3, we propose a solution exploiting the weak labels at training time only.

At training time, the Teacher network, trained on the pseudo-labels, is used to approximate the cost assignment function, allowing to match the best ordered set of weak labels with the tool instances. Once the set of eligible combinations is defined, this can be done by computing, for each combination, the complete set of permutations, and by selecting the ordered set of labels associated to the smallest cost, computed using the Teacher classifier. Beyond this interpretation, discussed in detail in Section 3.3, solving this problem can also be seen as finding the solution to a linear assignment problem for each combination set. Given N_{comb} combinations, this can be solved by applying N_{comb} times a Hungarian matching algorithm (Fig. A.2, B), and selecting the ordered set with the overall smallest cost, computed using the Teacher classifier.

Once the assignment between instances and weak labels is performed, the Student network is trained on the assigned weak labels. This extra step allows, at inference time, to directly predict instance classes from instance features, without requiring to access weak labels.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.media.2025.103471.

Data availability

Our method was evaluated on publicly available datasets.

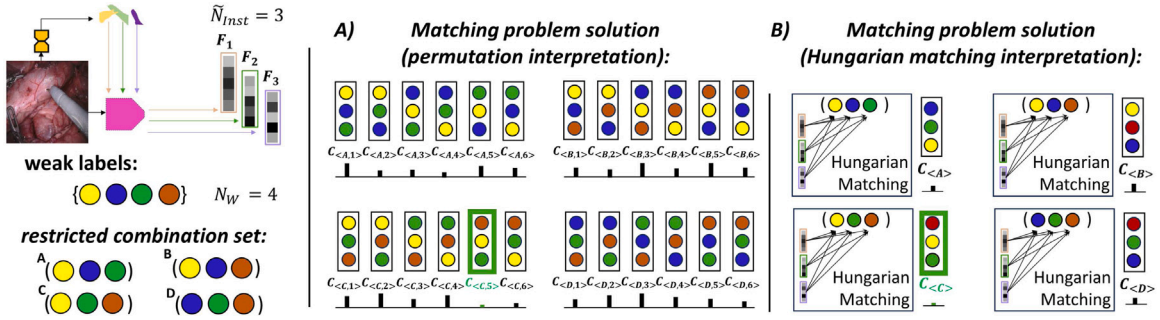

Fig. A.2. Matching problem solution interpretation. Given $\tilde{N}_{Inst}$ instances and the restricted combination set, obtained from the weak labels, a unique ordered set of labels, matched with the $\tilde{N}_{Inst}$ instances, must be identified. (A) Permutation interpretation of the matching problem solution: for each combination (A,B,C,D) the complete set of permutations (1,2,3,4,5,6) is computed. A cost gets assigned to each ordered set ($C_{\langle A,1\rangle}$, $C_{\langle A,2\rangle}$, $C_{\langle B,1\rangle}$, etc.). The ordered set with the smallest cost gets selected. (B) Hungarian matching interpretation of the matching problem solution: for each combination (A,B,C,D) the Hungarian matching algorithm identifies the ordered set with the smallest cost. Among the obtained ordered sets, the one with the smallest cost ($C_{\langle C\rangle}$) gets selected.

References

Allan, M., Kondo, S., Bodenstedt, S., Leger, S., Kadkhodamohammadi, R., Luengo, I., Fuentes, F., Flouty, E., Mohammed, A., Pedersen, M., et al., 2020. 2018 robotic scene segmentation challenge. arXiv preprint arXiv:2001.11190.

Allan, M., Ourselin, S., Hawkes, D.J., Kelly, J.D., Stoyanov, D., 2018. 3-D pose estimation of articulated instruments in robotic minimally invasive surgery. IEEE Trans. Med. Imaging 37 (5), 1204–1213.

Allan, M., Shvets, A., Kurmann, T., Zhang, Z., Duggal, R., Su, Y.-H., Rieke, N., Laina, I., Kalavakonda, N., Bodenstedt, S., et al., 2019. 2017 robotic instrument segmentation challenge. arXiv preprint arXiv:1902.06426.

Arthur, D., Vassilvitskii, S., 2006. k-means++: The Advantages of Careful Seeding. Technical Report, Stanford.

Bolelli, F., Allegretti, S., Baraldi, L., Grana, C., 2019. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. IEEE Trans. Image Process. 29, 1999–2012.

Bouget, D., Benenson, R., Omran, M., Riffaud, L., Schiele, B., Jannin, P., 2015. Detecting surgical tools by modelling local appearance and global shape. IEEE Trans. Med. Imaging 34 (12), 2603–2617.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. Springer, pp. 213–229.

da Costa Rocha, C., Padoy, N., Rosa, B., 2019. Self-supervised surgical tool segmentation using kinematic information. In: 2019 International Conference on Robotics and Automation. ICRA, IEEE, pp. 8720–8726.

Ding, H., Wu, J.Y., Li, Z., Unberath, M., 2023. Rethinking causality-driven robot tool segmentation with temporal constraints. Int. J. Comput. Assist. Radiol. Surg. 1–8.

Ding, H., Zhang, J., Kazanzides, P., Wu, J.Y., Unberath, M., 2022. Carts: Causality-driven robot tool segmentation from vision and kinematics data. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 387–398.

Durand, T., Mordan, T., Thome, N., Cord, M., 2017. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 642–651.

Francis, N., Curtis, N., Conti, J., Foster, J., Bonjer, H., Hanna, G., 2018. Eaes classification of intraoperative adverse events in laparoscopic surgery. Surg. Endosc. 32, 3822–3829.

Garcia-Peraza-Herrera, L.C., Li, W., Fidon, L., Gruijthuijsen, C., Devreker, A., Attilakos, G., Deprest, J., Vander Poorten, E., Stoyanov, D., Vercauteren, T., et al., 2017. Toolnet: holistically-nested real-time segmentation of robotic surgical tools. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 5717–5722.

González, C., Bravo-Sánchez, L., Arbelaez, P., 2020. Isinet: an instance-based approach for surgical instrument segmentation. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23. Springer, pp. 595–605.

Hasan, S.K., Linte, C.A., 2019. U-NetPlus: A modified encoder-decoder U-Net architecture for semantic and instance segmentation of surgical instruments from laparoscopic images. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society. EMBC, IEEE, pp. 7205–7211.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969.

Islam, M., Vibashan, V., Lim, C.M., Ren, H., 2021. ST-MTL: Spatio-temporal multitask learning model to predict scanpath while tracking instruments in robotic surgery. Med. Image Anal. 67, 101837.

Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F., 2020. A survey on contrastive self-supervised learning. Technol. 9 (1), 2.

Jin, Y., Cheng, K., Dou, Q., Heng, P.-A., 2019. Incorporating temporal prior from motion flow for instrument segmentation in minimally invasive surgery video. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 440–448.

Kalia, M., Aleef, T.A., Navab, N., Black, P., Salcudean, S.E., 2021. Co-generation and segmentation for generalized surgical instrument segmentation on unlabelled data. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 403–412.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised contrastive learning. Adv. Neural Inf. Process. Syst. 33, 18661–18673.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., et al., 2023. Segment anything. arXiv preprint arXiv:2304.02643.

Kong, X., Jin, Y., Dou, Q., Wang, Z., Wang, Z., Lu, B., Dong, E., Liu, Y.-H., Sun, D., 2021. Accurate instance segmentation of surgical instruments in robotic surgery: Model refinement and cross-dataset evaluation. Int. J. Comput. Assist. Radiol. Surg. 16 (9), 1607–1614.

Kurmann, T., Márquez-Neila, P., Allan, M., Wolf, S., Sznitman, R., 2021. Mask then classify: multi-instance segmentation for surgical instruments. Int. J. Comput. Assist. Radiol. Surg. 16 (7), 1227–1236.

Laina, I., Rieke, N., Rupprecht, C., Vizcaíno, J.P., Eslami, A., Tombari, F., Navab, N., 2017. Concurrent segmentation and localization for tracking of surgical instruments. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 664–672.

Lavanchy, J.L., Zindel, J., Kirtac, K., Twick, I., Hosgor, E., Candinas, D., Beldi, G., 2021. Automation of surgical skill assessment using a three-stage machine learning algorithm. Sci. Rep. 11 (1), 1–9.

Lee, E.-J., Plishker, W., Liu, X., Bhattacharyya, S.S., Shekhar, R., 2019. Weakly supervised segmentation for real-time surgical tool tracking. Heal. Technol. Lett. 6 (6), 231–236.

Liang, T., Glossner, J., Wang, L., Shi, S., Zhang, X., 2021. Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing 461, 370–403.

Liu, D., Wei, Y., Jiang, T., Wang, Y., Miao, R., Shan, F., Li, Z., 2020. Unsupervised surgical instrument segmentation via anchor generation and semantic diffusion. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 657–667.

Mascagni, P., Vardazaryan, A., Alapatt, D., Urade, T., Emre, T., Fiorillo, C., Pessaux, P., Mutter, D., Marescaux, J., Costamagna, G., et al., 2022. Artificial intelligence for surgical safety: automatic assessment of the critical view of safety in laparoscopic cholecystectomy using deep learning. Ann. Surg. 275 (5), 955–961.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T., 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4040–4048.

Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C., 2022. Trackformer: Multi-object tracking with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8844–8854.

Nwoye, C.I., Mutter, D., Marescaux, J., Padoy, N., 2019. Weakly supervised convolutional LSTM approach for tool tracking in laparoscopic videos. Int. J. Comput. Assist. Radiol. Surg. 14, 1059–1067.

Padoy, N., Blum, T., Ahmadi, S.-A., Feussner, H., Berger, M.-O., Navab, N., 2012. Statistical modeling and recognition of surgical workflow. Med. Image Anal. 16 (3), 632–641.

Pakhomov, D., Premachandran, V., Allan, M., Azizian, M., Navab, N., 2019. Deep residual learning for instrument segmentation in robotic surgery. In: International Workshop on Machine Learning in Medical Imaging. Springer, pp. 566–573.

Pakhomov, D., Shen, W., Navab, N., 2020. Towards unsupervised learning for instrument segmentation in robotic surgery with cycle-consistent adversarial networks. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 8499–8504.

Ramesh, S., Srivastav, V., Alapatt, D., Yu, T., Murali, A., Sestini, L., Nwoye, C.I., Hamoud, I., Fleurentin, A., Exarchakis, G., et al., 2022. Dissecting self-supervised learning methods for surgical computer vision. arXiv preprint arXiv:2207.00449.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Ross, T., Zimmerer, D., Vemuri, A., Isensee, F., Wiesenfarth, M., Bodenstedt, S., Both, F., Kessler, P., Wagner, M., Müller, B., et al., 2018. Exploiting the potential of unlabeled endoscopic video data with self-supervised learning. Int. J. Comput. Assist. Radiol. Surg. 13 (6), 925–933.

Russo, M., 2012. Triangulation concept for minimally invasive access surgery. US Patent App. 13/442, 006.

Sahu, M., Strömsdörfer, R., Mukhopadhyay, A., Zachow, S., 2020. Endo-Sim2Real: Consistency learning-based domain adaptation for instrument segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 784–794.

Sestini, L., Rosa, B., De Momi, E., Ferrigno, G., Padoy, N., 2021. A kinematic bottleneck approach for pose regression of flexible surgical instruments directly from images. IEEE Robot. Autom. Lett. 6 (2), 2938–2945.

Sestini, L., Rosa, B., De Momi, E., Ferrigno, G., Padoy, N., 2023. FUN-SIS: A fully unsupervised approach for surgical instrument segmentation. Med. Image Anal. 102751.

Shvets, A.A., Rakhlin, A., Kalinin, A.A., Iglovikov, V.I., 2018. Automatic instrument segmentation in robot-assisted surgery using deep learning. In: 2018 17th IEEE International Conference on Machine Learning and Applications. ICMLA, IEEE, pp. 624–628.

Tanzi, L., Piazzolla, P., Porpiglia, F., Vezzetti, E., 2021. Real-time deep learning semantic segmentation during intra-operative surgery for 3D augmented reality assistance. Int. J. Comput. Assist. Radiol. Surg. 16 (9), 1435–1445.

Teed, Z., Deng, J., 2020. Raft: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision. Springer, pp. 402–419.

Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., De Mathelin, M., Padoy, N., 2016. EndoNet: a deep architecture for recognition tasks on laparoscopic videos. IEEE Trans. Med. Imaging 36 (1), 86–97.

Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9 (11).

Vardazaryan, A., Mutter, D., Marescaux, J., Padoy, N., 2018. Weakly-supervised learning for tool localization in laparoscopic videos. In: Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis: 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3. Springer, pp. 169–179.

Voros, S., Long, J.-A., Cinquin, P., 2006. Automatic localization of laparoscopic instruments for the visual servoing of an endoscopic camera holder. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006: 9th International Conference, Copenhagen, Denmark, October 1-6, 2006. Proceedings, Part I 9. Springer, pp. 535–542.

Wang, Y., Long, Y., Fan, S.H., Dou, Q., 2022. Neural rendering for stereo 3d reconstruction of deformable tissues in robotic surgery. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 431–441.

Wei, G.-Q., Arbter, K., Hirzinger, G., 1997. Automatic tracking of laparoscopic instruments by color coding. In: CVRMed-MRCAS'97: First Joint Conference Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery Grenoble, France, March 19–22, 1997 Proceedings. Springer, pp. 357–366.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. Adv. Neural Inf. Process. Syst. 34, 12077–12090.

Xue, Y., Liu, S., Li, Y., Wang, P., Qian, X., 2022. A new weakly supervised strategy for surgical tool detection. Knowl.-Based Syst. 239, 107860.

Yang, Z., Simon, R., Linte, C., 2022. A weakly supervised learning approach for surgical instrument segmentation from laparoscopic video sequences. In: Medical Imaging 2022: Image-Guided Procedures, Robotic Interventions, and Modeling. Vol. 12034, SPIE, pp. 412–417.

Zhao, Z., Jin, Y., Gao, X., Dou, Q., Heng, P.-A., 2020. Learning motion flows for semi-supervised instrument segmentation from robotic surgical video. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 679–689.

Zhao, Z., Jin, Y., Heng, P.-A., 2022. Trasetr: track-to-segment transformer with contrastive query for instance-level instrument segmentation in robotic surgery. In: 2022 International Conference on Robotics and Automation. ICRA, IEEE, pp. 11186–11193.

Zia, A., Bhattacharyya, K., Liu, X., Berniker, M., Wang, Z., Nespolo, R., Kondo, S., Kasai, S., Hirasawa, K., Liu, B., et al., 2023. Surgical tool classification and localization: results and methods from the MICCAI 2022 SurgToolLoc challenge. arXiv preprint arXiv:2305.07152.