Neural Networks 200 (2026) 108808

Contents lists available at ScienceDirect

Neural Networks

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

Full Length Article

Few-shot video object segmentation in X-ray angiography using local matching and spatio-temporal consistency loss

Lin Xi [a], Yingliang Ma [a,b,*], Xiahai Zhuang [c]

[a] University of East Anglia, United Kingdom
[b] King's College London, United Kingdom
[c] Fudan University, China

ARTICLE INFO

Keywords:
X-ray video segmentation
Few-shot video object segmentation
Spatio-temporal consistency
Medical image dataset

ABSTRACT

High-quality, densely annotated data serve as a crucial foundation for developing robust X-ray angiography segmentation models. However, obtaining per-object pixel-level annotations in the medical domain is both expensive and time-consuming, often requiring close collaboration between clinical experts and developers. This paper aims to reduce the annotation costs of X-ray angiography videos by leveraging few-shot video object segmentation (FSVOS), which separates target objects from the background using only a single annotated frame during inference. We introduce a novel FSVOS model that employs a local matching strategy to restrict the search space to the most relevant neighboring pixels. Rather than relying on inefficient standard im2col-like implementations (*e.g.*, spatial convolutions, depthwise convolutions and feature-shifting mechanisms) or hardware-specific CUDA kernels (*e.g.*, deformable and neighborhood attention), which often suffer from limited portability across non-CUDA devices, we reorganize the local sampling process through a direction-based sampling perspective. Specifically, we implement a non-parametric sampling mechanism that enables dynamically varying sampling regions. This approach provides the flexibility to adapt to diverse spatial structures without the computational costs of parametric layers and the need for model retraining. To further enhance feature coherence across frames, we design a supervised spatio-temporal contrastive learning scheme that enforces consistency in feature representations. In addition, we introduce a publicly available benchmark dataset for multi-object segmentation in X-ray angiography videos (MOSXAV), featuring detailed, manually labeled segmentation ground truth. Extensive experiments on the CADICA, XACV, and MOSXAV datasets show that our proposed FSVOS method outperforms current state-of-the-art video segmentation methods in terms of segmentation accuracy and generalization capability (*i.e.*, seen and unseen categories). This work offers enhanced flexibility and potential for a wide range of clinical applications. Code is available at: https://github.com/xilin-x/XRAVOS.

1. Introduction

X-ray angiography video segmentation of blood vessels and other surgical objects is a critical step in 3D vessel reconstruction, coronary artery analysis, and cardiac modeling. Frame-wise and pixel-wise segmentation enables precise localization of each object in X-ray videos, aiding in surgical planning, biomarker computation, and various downstream tasks. However, in many practical scenarios, obtaining dense annotations from clinical experts, especially pixel-level annotations for each frame, is labor-intensive and time-consuming. Moreover, the limited availability of samples for rare anomalies and unusual pathological conditions further complicates the annotation process.

Driven by the paradigm of class-agnostic mask tracking, the computer vision community has increasingly focused on few-shot video object segmentation (FSVOS) (Caelles et al., 2017; Cheng & Schwing, 2022; Cheng et al., 2021b; Gu et al., 2019; Oh et al., 2019; Perazzi et al., 2017; Zheng et al., 2024). This shift prioritizes robust tracking mechanisms over the explicit learning of object-specific semantic representations, thereby mitigating the heavy reliance on large-scale annotated datasets. The goal of FSVOS is to segment target objects throughout an entire video sequence based on an initial object mask, which can be either manually provided or automatically generated from any single frame. As a widely accepted solution for FSVOS, matching-based methods (Cheng & Schwing, 2022; Cheng et al., 2021b; Oh et al., 2019) perform a matching strategy to generate dense correspondences between the query frame and the support frame(s), where the past frames as well as corresponding masks are stored in an external memory to build a feature memory (Yang et al., 2021b). In Space-Time Memory (STM) network

* Corresponding authors.
E-mail addresses: xilin.chibchin@outlook.com (L. Xi), yingliang.ma@uea.ac.uk (Y. Ma), zxh@fudan.edu.cn (X. Zhuang).

https://doi.org/10.1016/j.neunet.2026.108808
Received 25 August 2025; Received in revised form 14 February 2026; Accepted 2 March 2026
Available online 7 March 2026

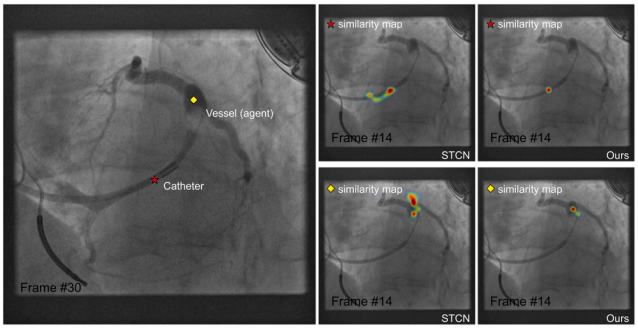

Fig. 1. Visualization of similarity maps between the query frame (#30) and the support frame (#14). Reference labels manually selected in the query frame (leftmost) are used to compute similarity scores relative to the support frame. In the resulting heatmaps, regions in deep red indicate higher similarity. The yellow diamond (◆) denotes the coronary vessels, while the red star (★) identifies the injection catheter.

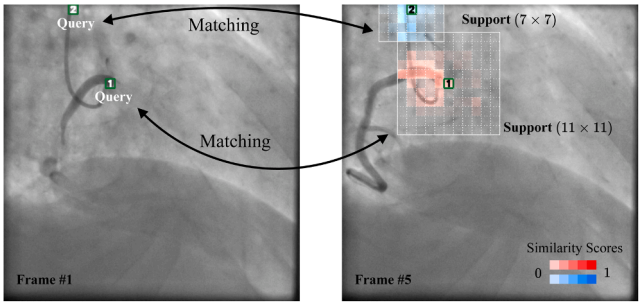

Fig. 2. An illustration of our local matching between the query frame and the support frame at the given patch (position **1** and **2**). Position 1 and 2 on the key frames correspond to the spatial positions of 1 and 2 in the current frame. The similarity scores are represented by colorful heatmap masks (*i.e.*, ■ and ■), where darker colors indicate higher similarity.

(Oh et al., 2019), cross-image correspondences is the key component that makes it superior in performance and simplicity. In the pixel-wise retrieval phase, it enables the global content-dependent interactions among different image regions to model long-range dependencies. Through the visualization of dense space-time correspondence, we observe the crucial role of local contexts, such as the neighborhood pixels (Hassani et al., 2023; Liu et al., 2021; Pan et al., 2023; Ramachandran et al., 2019; Yang et al., 2021a), in establishing correspondences for cross-frame matching (see the similarity scores in Fig. 1). Meanwhile, compared to local-side computing solutions (Zhang et al., 2025), global and fine-grained dependencies become less critical due to spatial redundancy and noise as the number of features involved in non-local matching increases, particularly in cases of locally recurring motion such as cardiac and respiratory movements, or within specific regions of interest in X-ray angiography videos. On the other hand, matching-based methods (Cheng & Schwing, 2022; Cheng et al., 2021b; Liu et al., 2025; Oh et al., 2019) focus only on discovering space-time correspondences from inter-frames, while ignoring the valuable temporal consistency of object representation across multiple frames. Indeed, due to only mining the matched pixels across the memory frames, some background pixels/patches are wrongly recognized as highly correlated to the query primary objects. Therefore, it is necessary to make full use of inter-frame consistency to make up for the drawbacks caused by ignoring temporal coherence.

In the domain of vision Transformers, window-based self-attention (Liu et al., 2021; Yang et al., 2021a) has emerged as a powerful and efficient mechanism for capturing local dependencies. The Swin Transformer (Liu et al., 2021) partitions feature maps into non-overlapping windows and applies self-attention to each independently, leveraging batched matrix multiplication for high parallelization. Similarly, the Focal Transformer (Yang et al., 2021a) employs fine-grained local attention to account for short-range visual dependencies. These architectural advancements provide a strong motivation to model localized visual dependencies as a means of mitigating the high computational costs inherent in existing global matching-based methods (Cheng & Schwing, 2022; Cheng et al., 2021b; Liu et al., 2025; Oh et al., 2019). Current paradigms based on the matching framework generate global correlations to facilitate segmentation. While effective, a significant limitation of these methods is that fine-level local interactions between the query and the memory become increasingly diluted as the memory bank grows. This accumulation of memory pixels introduces noise and reduces the signal-to-noise ratio, ultimately limiting the model's capacity to extract highly accurate and well-localized correspondences. A parallel challenge exists in standard Transformer architectures, where global self- and cross-attention mechanisms often struggle to preserve fine-grained spatial details when processing large feature maps. Inspired by local attention mechanisms (Hassani et al., 2023; Pan et al., 2023; Ramachandran et al., 2019), we explore local correspondences within a relevant neighborhood of sampling locations to enhance the robustness of feature matching. Furthermore, to improve feature coherence, we reinforce spatio-temporal correspondences by enforcing globally consistent feature representations across frames, ensuring that local precision does not come at the expense of global context.

In this work, we propose a novel FSVOS framework for X-ray angiography videos that captures local correspondences during the feature retrieval process between query and support frames. Since visual dependencies are typically stronger among nearby regions than distant ones, we restrict the correspondence search to a neighborhood of sampling locations in the support frames, as depicted in Fig. 2. By focusing on relevant local regions rather than the global field, this design improves the robustness of fine-grained matching while maintaining a low computational cost. Each query feature attends to its closest neighborhood in the support frames at the same granularity, effectively covering the most relevant regions while substantially reducing the number of spatio-temporal computations compared to dense matching mechanisms such as (Cheng & Schwing, 2022; Cheng et al., 2021b). Moreover, as shown in Fig. 1, global matching strategies often attend to uninformative background regions, causing salient foreground objects to be submerged in interference signals. From an implementation prespective, existing local-attention and matching approaches suffer from notable practical limitations. On one hand, methods based on standard im2col-like implementations (Liu et al., 2021; Pan et al., 2023; Ramachandran et al., 2019; Yang et al., 2021a) exhibit sub-optimal memory efficiency and inefficient data access patterns. Specifically, Stand-Alone Self-Attention (SASA) (Ramachandran et al., 2019) and Slide Attention (Pan et al., 2023) are typically implemented using generic spatial or depth-wise convolutions. These designs incur redundant costs in kernel parameter storage and loading, leading to inefficient memory utilization. Similarly, Swin Transformer (Liu et al., 2021) and Focal Transformer (Yang et al., 2021a) adopt window-based partitioning with cyclic shifts executed through reshape and roll operations. While effective for batch training, these shifts introduce non-adjacent spatial regions into the same attention window, requiring specialized attention masks to preserve local receptive fields. As a result, substantial intermediate data movement and increased memory consumption are often unavoidable. In contrast, our framework leverages a non-parametric sampling operation that avoids these architectural inefficiencies and hardware dependencies. This design ensures no additional inference-time overhead compared to existing matching-based models, facilitating seamless deployment across diverse clinical computing environments, including the CPU-only systems commonly found in hospital settings. To further enforce spatio-temporal consistency, we incorporate a supervised contrastive learning scheme during traning. By defining object centers based on ground-truth labels, we encourage pixels belonging to the same object to form compact

feature clusters while maintaining clear separation from other objects. This framework explicitly links query-frame features with cross-frame global representations. Crucially, this scheme requires no additional pixel-wise annotations and incurs no extra cost during inference, making the overall framework both data-efficient and computationally practical.

We validate the effectiveness of our method by evaluating it on various public X-ray angiography benchmark video datasets, *i.e.*, CADICA (Jiménez-Partinen et al., 2024), XACV (Wu et al., 2025), and MOSXAV. On all datasets, our method achieves a competitive performance compared to the current state-of-the-art methods while maintaining a real-time inference speed. In particular, instead of a global matching mechanism, our approach uses local matching to effectively explore pixel dependencies compared with previous matching-based solutions (Cheng & Schwing, 2022; Cheng et al., 2021b; Oh et al., 2019) in the X-ray angiography video segmentation. Note that our all our experiments are implemented on a single NVIDIA RTX™ 6000 GPU and CPU platform, which makes our method much more accessible than other state-of-the-art methods, which require powerful hardware resources. Our main contributions can be summarized as follows:

- We propose a novel FSVOS method for X-ray angiography video by exploiting the most relevant neighborhood region.
- To make the sampling process simpler and more flexible, we propose a non-parametric sampling method that enables dynamically adjustable sampling regions without requiring retraining.
- We introduce a new benchmark dataset[1], the X-ray Multi-Object Segmentation in X-ray Angiography Videos (MOSXAV). MOSXAV focuses on understanding multiple objects in X-ray videos, offering high-quality, manually labeled segmentation ground truth.

2. Related works

2.1. Vessel and catheter segmentation

Vessel segmentation is an active research topic in medical image analysis. Among the most widely used backbones, U-Net (Ronneberger et al., 2015) has recently become the most popular. To address the limitation of U-Net, which only allows a single set of concatenation layers between encoder and decoder blocks, T-Net (Jun et al., 2020) introduces a nested encoder-decoder architecture. This design employs multiple small encoder-decoder sub-networks for each block of the convolutional network, enabling more effective vessel segmentation in coronary angiography. Sequential-image-based approaches, such as SVS-Net (Hao et al., 2020), adopt an encoder-decoder architecture that leverages multiple contextual frames of 2D images to improve segmentation of 2D vessel masks. Khan et al. (Khan et al., 2023) proposed a contextual network for accurate retinal vessel identification, with particular consideration of computational efficiency to support deployment on resource-constrained devices such as smartphones. Recent methods have also integrated ViT (Dosovitskiy et al., 2021) with graph neural networks or convolutional block attention modules to improve structural understanding of vascular morphology, as exemplified by G2ViT (Xu & Wu, 2024) and CBAM&ViT (Ma et al., 2025).

In addition, automatic catheter segmentation has emerged as another active topic in medical image analysis. Existing work has reported catheter segmentation in X-ray images, including electrophysiology (EP) electrodes (Baur et al., 2016) and EP catheters (Ambrosini et al., 2017; Wu et al., 2015; Xi et al., 2025). More recently, Kim et al. (Kim et al., 2022) addressed the challenge of detecting active acoustic catheters (AACs) in echocardiography. Their method first applies U-Net to segment the left ventricle, then filters the color response generated by the AAC, and finally performs thresholding to identify the catheter. Yang et al. (Yang et al., 2019) proposed a 3D ultrasound catheter segmentation method based on Shared-ConvNet, which employs CNNs with shared weights across imaging planes and performs voxel-wise binary classification. Overall, beyond our prior work, the literature on catheter segmentation in AI-driven, ultrasound-guided minimally invasive endovascular surgery (MIES) remains limited. Nguyen et al. (Nguyen et al., 2020) introduced an end-to-end, real-time deep learning framework for endovascular intervention, incorporating a novel flow-guided warping function to enforce frame-to-frame temporal continuity. To capture such sequential behavior, Ranne et al. (Ranne et al., 2024) proposed the Attention-in-Attention ResNet for Segmentation (AiAReSeg), which aggregates information across image sequences to infer knowledge about the current frame.

2.2. Few-shot video object segmentation

FSVOS (Caelles et al., 2017) differs from fully automated video segmentation (Lu et al., 2019; Xi et al., 2022, 2024) by incorporating human input, specifically a first-frame segmentation mask of the target objects. The methods then segment the desired objects in the remaining frames. Here, few-shot refers to the level of human interaction at test time, not during the training phase. The few-shot methods still rely on the supervised learning paradigm to train a pixel-wise tracking (Cheng & Schwing, 2022; Cheng et al., 2021b; Liu et al., 2025; Oh et al., 2019) or mask propagation framework. Early few-shot methods (Caelles et al., 2017; Maninis et al., 2019) employ online fine-tuning on the basis of this supervision, which suffered from high test runtime and were gradually phased out. The follow-up researches have been explored including embedding learning (Voigtlaender et al., 2019; Yang et al., 2020), propagation-based (Perazzi et al., 2017; Ventura et al., 2019) and tracking-based (Hu et al., 2017; Wang et al., 2019). These methods perform frame-to-frame propagation or global matching with the first reference frame, while the context is still limited and it becomes harder to match as the video progresses. To fully exploit the spatial-temporal context over longer distances, recent state-of-the-art methods define the past frames with object masks as feature memory and the current frame as the query (Oh et al., 2019; Robinson et al., 2020). As a representative work, STM (Oh et al., 2019) utilizes an external memory to store the past frames as well as corresponding masks, where a dense matching is adopted to establish long-range dependencies in order to achieve pixel classification and objects segmentation. Staring from STM (Oh et al., 2019), matching-based paradigm was adopted and has been extended by many follow-up works (Cheng & Schwing, 2022; Cheng et al., 2021a,b; Liu et al., 2025), which achieve leading accuracy on most benchmarks duo to long-range context support.

The latest research in matching-based FSVOS has primarily concentrated on enhancing network architectures, such as STCN (Cheng et al., 2021b), XMem (Cheng & Schwing, 2022), and LiVOS (Liu et al., 2025). STCN (Cheng et al., 2021b) employs an effective and efficient matching strategy for establishing dense correspondences between query and support frame(s), eliminating the need for re-encoding the mask features for every objects, as observed in STM (Oh et al., 2019). These methods often prioritize the exploration of valid global correspondences while disregarding the shaping of the discriminative embedding space. Moreover, our contribution builds upon these studies, as we have not only developed a more efficient matching model based on the current matching-based FSVOS methods (Cheng & Schwing, 2022; Cheng et al., 2021b; Liu et al., 2025) but also made advancements in ensuring consistency between query and memory in the aspect of feature learning.

In medical data analysis, there has been increasing interest in FSVOS techniques that enable class-agnostic mask tracking and segmentation using only a few annotated examples. For instance, RAB (Zheng et al., 2024) is a two-phase framework that introduces a spatio-temporal consistency relearning strategy to adapt an image segmentation model for video data. In contrast, our approach is a one-stage, matching-based method that not only develops a more efficient matching strategy but also strengthens the consistency between query and support frames through improved feature learning.

[1] https://github.com/xilin-x/MOSXAV

2.3. Transformer architectures

Transformer architectures are a type of neural network architecture that have proven extremely adept at modelling long-term relationships within an input sequence via self-attention mechanisms (Vaswani et al., 2017). The exploration of global matching between query and memory in STM network and their variants closely resembles the self/cross-attention mechanisms found in Transformer. This exploration aims to uncover global correlations. Balancing interaction granularity is a persistent challenge for Transformer-based methods. Because of the significant computational burden associated with global interaction, the input features are often downsampled to a lower resolution, which to some extent restricts the networks' capability for fine-grained feature learning.

Recently, many efforts have been made to address this problem, which can be roughly divided into two categories in practice. The first category involves employing local attention modules to alleviate the issues mentioned above. The most direct approach is constraining the attention pattern to fixed local windows, which is commonly adopted by many works (Chen et al., 2022; Dosovitskiy et al., 2021; Hassani et al., 2023; Liu et al., 2021; Yang et al., 2021a). While restricting the attention pattern to a local neighborhood can reduce complexity, it also sacrifices global information. The second category is to learn data-dependent sparse attention. Inspired by deformable convolution (Dai et al., 2017), Deformable DETR (Zhu et al., 2021) directs its attention to a small fixed set of sampling points, predicted from the feature of query elements. PnP-DETR (Wang et al., 2021) presents a similar idea, where it samples fine foreground features and pools background features into a reduce size. Sparse DETR (Roh et al., 2022) sparsifies encoder tokens by using a learnable decoder attention map predictor, which is different from Deformable DETR's key sparsification method.

In our method, a local matching module is proposed to enable effective message passing between the query and support frame(s) within a limited computational budget, focusing on fine-grained, locally relevant regions. Similar to local attention mechanisms in transformers (Hassani et al., 2023; Pan et al., 2023; Ramachandran et al., 2019), the query captures visual dependencies for specific locations within its local neighborhood; however, in our case, these dependencies are established with the corresponding regions of the support frame(s). To further improve flexibility and efficiency, we employ a non-parametric sampling strategy that allows dynamically varying sampling regions without requiring retraining or device-specific support.

2.4. Supervised contrastive learning

Supervised contrastive learning draws on existing literature in self-supervised representation learning, metric learning, and supervised learning. Typically, the state-of-the-art family of models for self-supervised representation learning utilizes the paradigm known as contrastive learning (Chen et al., 2020; Hénaff, 2020; Hjelm et al., 2019; Sermanet et al., 2018; Tian et al., 2020; Tschannen et al., 2020; Wu et al., 2018). In these works, the losses are inspired by noise contrastive estimation (Gutmann & Hyvärinen, 2010; Mnih & Kavukcuoglu, 2013) or N-pair losses (Sohn, 2016). Recently, supervised contrastive loss (Khosla et al., 2020) has been introduced as a novel extension to the contrastive loss function. This innovative approach allows for multiple positives per anchor, thus adapting contrastive learning to the fully supervised setting. Following (Khosla et al., 2020), supervised contrastive loss has also been employed in various downstream tasks (Zhang et al., 2021; Zhao et al., 2021). To enhance consistency between query and memory, we proposed a supervised contrastive learning scheme. In contrast to the traditional supervised contrastive loss (Khosla et al., 2020), which does not explicitly enforce intra-class compactness, our approach defines object centers based on ground-truth labels, encouraging pixels of the same object to form compact feature clusters while preserving clear separation across different objects.

3. Method

3.1. Network formulation

To modeling space-time correspondences in the context of X-ray video, our model architecture shares a similar multi-store feature memory design with (Cheng et al., 2021b; Oh et al., 2019). The overall framework of the proposed method is shown in Fig. 3.

Given a video with T frames $\mathcal{I} = \{\boldsymbol{I}_t\}_1^T$ and first-frame reference mask $\boldsymbol{M}_1$, our model match the object pixels and generates corresponding masks for subsequent query frames. For the current frame $\boldsymbol{I}_t$, we first fed it into the query encoder to extract the query feature $\boldsymbol{Q} \in \mathbb{R}^{HW \times D}$ in which the two spatial dimension were flattened. Subsequently, the spatial feature map $\boldsymbol{Q}$ is passed to the affinity module, where an affinity matrix $\boldsymbol{A}$ is obtained by applying a softmax function on a similarity matrix $\boldsymbol{S}$ between $\boldsymbol{Q}$ and the key feature $\boldsymbol{K} \in \mathbb{R}^{KNHW \times D}$, where K is the number of the keyframes and N is the number of local sampling elements. The similarity matrix $\boldsymbol{S}$ captures the pairwise similarities between each sampled key element (*i.e.*, the most relevant neighborhood region) and every query element. Here, the local elements in the sampled key are generated through the proposed local sampling module, which samples key and value feature elements in parallel before storing them in memory. After obtaining the affinity $\boldsymbol{A}$, we retrieve the corresponding feature $\boldsymbol{F} \in \mathbb{R}^{HW \times C}$ from the external feature memory through memory reading, and then the readout feature $\boldsymbol{F}$ are employed in the generation of a segmentation mask for the current frame. In particular, the new memory elements which are generated by our proposed local sampling module can be added to the external memory when if the current frame is the keyframe. For the keyframe, we will take a fixed interval sample frame every rth frame and identify it to keyframe in practice.

For the local sampling module, the inputs consist of the key and value features extracted from both the query and the value encoder. Within this module, fine-grained features are obtained from the corresponding neighborhood region of the keyframes' features according to specific locations of query element. The local sampling mechanism gathers tokens from both the key and value features in a consistent manner, thereby aggregating local contextual information. In contrast, standard STM network are designed to capture long-range dependencies at a fine-grained level, but they incur high computational costs when performing attention between the query and a large memory set. To address this limitation, we propose a local matching strategy that reduces the computational burden while preserving representational effectiveness.

3.2. Local matching and aggregation

The core idea of our matching and aggregation is to find a correlation matrix $\boldsymbol{C}$ between the current frame and the history to retrieve the corresponding features from the memory, which is more efficient and effective to establish dense correspondences for matching-based matching solution. Instead of attending to all key elements at a fine-grained level during matching, we proposed attending only to the most relevant neighborhood fine-grained key elements locally. As such, it can use as many sizes of the memory bank as the standard matching-based model but with much longer history coverage.

3.2.1. Matching

Consider a query $\boldsymbol{Q} \in \mathbb{R}^{HW \times D}$ and the keys $\boldsymbol{K} \in \mathbb{R}^{KHW \times D}$, an ith query element is associated with a jth key element in the memory, where K is the number of the keyframes, *i.e.*, $KHW = K \times H \times W$ — we refer to them jointly $\mathcal{M} = \{(i, j), c_{ij}\} \subset \boldsymbol{Q} \times \boldsymbol{K}$ as the feature matches. The i and j index a query element $\boldsymbol{q}_i$ and a key element $\boldsymbol{k}_j$, as well as a similarity score $\boldsymbol{s}_{ij}$.

As in the standard matching-based matching, the assignment $\mathcal{M}$ can be obtained by computing the correlation matrix $\boldsymbol{C} = \{c_{ij}; i \in \Lambda, j \in \Omega\}$ for all possible matches, where Λ and Ω specify the set of query and key elements, respectively. Meanwhile, instead of attending to all key/value

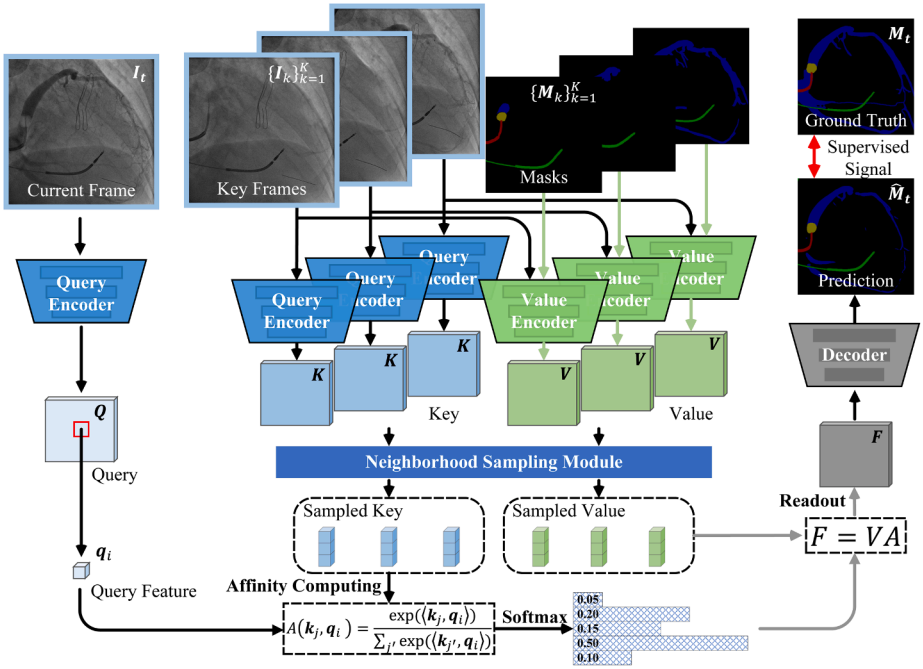

Fig. 3. Overview of our proposed local matching-based FSVOS method. Given a query frame $\boldsymbol{I}_t$ and several key frames $\{\boldsymbol{I}_k\}_{k=1}^K$, sampled from a same training video, we first use local sampling function $\phi(\cdot;\cdot)$ to retrieve local sampled set Ω_L from key features $\boldsymbol{K}$ and value features $\boldsymbol{V}$. Then we make feature-level correspondence matching between query feature $\boldsymbol{Q}$ and sampled key features through Eqs. (2)–(3). Then we retrieve corresponding features $\boldsymbol{F}$ by aggregating sampled value features using the affinity matrix $\boldsymbol{A}$.

elements in the key/value sampled set Ω for a specific query element i, we sample a total of N elements from the keyframes (support) to construct the key $\boldsymbol{K}$ and the value $\boldsymbol{V}$,

$$\Omega = \Omega_L, \tag{1}$$

where Ω_L denotes the local sampled set, and $KN < KHW$. In our case, the matching-based FSVOS is equivalent to solving a neighborhood voting problem, such as Nearest Neighbor Search (NNS) (Indyk & Motwani, 1998), which aggregate the matched representations by a similarity matrix $\boldsymbol{S} \in \mathbb{R}^{N\times HW}$ representing the weights.

3.2.2. Affinity computing

For clarity, we consider a single keyframe (*i.e.*, $K=1$). The affinity matrix $\boldsymbol{A}$ is derived by applying a softmax operation along the memory dimension (rows) of the similarity matrix $\boldsymbol{S}$:

$$\boldsymbol{A}(\boldsymbol{k}_j, \boldsymbol{q}_i) = \mathrm{softmax}(\boldsymbol{S}(\boldsymbol{k}_j, \boldsymbol{q}_i)), \tag{2}$$

where i and j denote the index of the query element and the key element. The similarity matrix is computed by

$$\boldsymbol{S}(\boldsymbol{k}_j, \boldsymbol{q}_i) = \langle \boldsymbol{k}_j, \boldsymbol{q}_i \rangle, \tag{3}$$

where $\langle\cdot,\cdot\rangle$ denotes a similarity measure, *i.e.*, ℓ_2 distance. Given the ith query element $\boldsymbol{q}_i$ obtained from the query frame via the query encoder, the sampled key elements $\{\boldsymbol{k}_j; j \in \Omega\}$ are collected from the local set, which is derive from the memory. The total number of key elements for the ith query element is N.

3.2.3. Local sampling

To efficiently search for the most relevant neighborhood region, we proposed a *local sampling module*. Similar to the local attention mechanism in ViT's variants, we retrieve a small relevant subset $\Omega_L \subset \Omega$ from all spatial locations as the local set. Specifically, we define the sampling function $\phi(\cdot;\cdot)$ to attends only to a small set of sampling points around a reference point, independent of the spatial size of the feature maps, as shown in Fig. 4. By assigning a fixed, limited number of keys to each

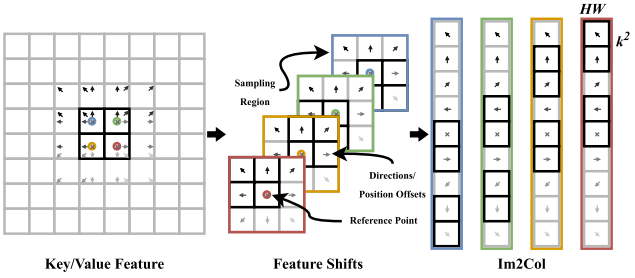

Fig. 4. An illustration of our local sampling operation at feature level.

query, this approach helps mitigate issues related to convergence and feature spatial resolution.

To achieve efficient affinity computation, the sampled local set needs to be packed into a matrix to enable parallel processing on accelerators. For that, the im2col operation (Chellapilla et al., 2006; Jia et al., 2014) reorganizes image patches into columns of a matrix to implement general matrix multiply (GEMM), which can be efficiently executed using optimized basic linear algebra subprograms (BLAS) libraries. As depicted in Fig. 4, the sampling region is centered at a specific query position (*i.e.*, the reference point in Fig. 4) and represents the region corresponding to its key/value pairs in memory. The sampled set is then flattened into columns, forming the final key/value matrix $\boldsymbol{M}$, which can be expressed as:

$$\begin{aligned} &\boldsymbol{M}[i,j] = \phi(\mathrm{p}(i), \Delta\mathrm{p}(j)) = \boldsymbol{X}[\mathrm{p}(i) + \Delta\mathrm{p}(j)], \\ &\text{with } i \in [0, HW-1] \text{ and } j \in [0, k^2-1] \end{aligned} \tag{4}$$

where $\mathrm{p}(\cdot)$ represents the position function, and $\Delta\mathrm{p}(\cdot)$ is sampling direction (*i.e.*, position offsets) determined by a regular grid $\mathcal{R}$ over the input feature map $\boldsymbol{X}$, where $\boldsymbol{X} = \{\boldsymbol{K}, \boldsymbol{V}\}$. The grid $\mathcal{R}$ defines the neighborhood region; for instance, when $k=3$, it is given by:

$$\mathcal{R} = \{(-1,-1), (0,-1), \ldots, (0,1), (1,1)\} \tag{5}$$

which corresponds to a 3×3 region with 9 possible directions. Notably, the sampling process primarily involves independently slicing the feature map according to the sampling windows from the sampling reference point perspective, as illustrated by the colored windows in Fig. 4, similar to the approach used in SASA (Ramachandran et al., 2019) and NAT (Hassani et al., 2023).

However, the above im2col operation can also be reinterpreted from a sampling direction-based perspective. In this view, the different directions, as illustrated in Fig. 4, determine the structure of the key/value matrix, which contains k^2 rows. Each row corresponds to shifting the feature map in a specific direction, allowing us to reformulate the above equations as:

$$\boldsymbol{M}[:, j] = \phi(:, \Delta \mathrm{p}(j)), \forall i, \tag{6}$$

which is equivalent to shifting the original feature map in a specific direction defined by $\mathcal{R}$.

Instead of using depth-wise convolution with predefined fixed kernels as a substitute for feature shifts, we directly apply the non-parametric unfold operation, which enables dynamically varying sampling regions without requiring retraining. In the depth-wise convolution-based approach, the learnable fixed-shaped weights and biases are stored as model parameters during training. In contrast, our proposed unfold-based solution leverages a non-parametric sampling process, allowing it to seamlessly adapt to arbitrary sampling regions. Eq. (6) becomes

$$\begin{aligned} \phi(i, \Delta \mathrm{p}(j)) &= \mathrm{unfold}(:, \Delta \mathrm{p}(j)) \\ &= \boldsymbol{X}[\mathrm{p}(i) + \Delta \mathrm{p}(j)], \forall i, j. \end{aligned} \tag{7}$$

In general, an efficient local sampling can be implemented from a direction-based perspective by carefully defining the sampling directions $\mathcal{R}$ within the unfold operation. This approach reduces the main computational overhead by avoiding inefficient slicing operations and leveraging optimized unfold operations on diverse hardware platforms (Chellapilla et al., 2006; Chetlur et al., 2014).

Finally, to identify the most relevant neighborhood region for each query element, it is necessary to determine the appropriate sampling reference position(s) $\mathrm{p}(i)$ in Eq. (7). In Focal Transformer (Yang et al., 2021a) and Slide Transformer (Pan et al., 2023), the query and key features are extracted from identical spatial locations within the same image space. Consequently, these methods directly use the spatial position of the query element as the sampling reference point, *i.e.*, $\mathrm{p}(i) = i$, where i denotes the spatial index of the query feature map. Under this formulation, each query element attends to its corresponding local neighborhood region in the keyframe feature map. However, considering that video frames often exhibit spatial dynamics, including both rigid and non-rigid motion, and that no explicit supervision is available during training, we further enhance the flexibility of the sampling process by incorporating the notion of cross-image similarity, as proposed in (Xie et al., 2022). Specifically, the sampling reference point is determined based on appearance similarity between object representations across frames, enabling more adaptive and robust neighborhood aggregation. Given a query element $\boldsymbol{q}_i$, we compute its feature similarity with key elements $\{\boldsymbol{k}_j; j \in \Omega\}$ using Eq. (3) and identify the key element $\boldsymbol{k}_{j^*}$ that yields the highest similarity score:

$$j^* = \arg\max_{j \in \Omega} \boldsymbol{S}(\boldsymbol{k}_j, \boldsymbol{q}_i). \tag{8}$$

The sampling reference point $\mathrm{p}(i)$ is then defined as the spatial position of the key element $\boldsymbol{k}_{j^*}$ that yields the highest similarity score:

$$\mathrm{p}(i) = \mathrm{pos}(\boldsymbol{k}_{j^*}), \tag{9}$$

where $\mathrm{pos}(\cdot)$ denotes the spatial position of the key element in the feature map, and the similarity scores are filtered using a top-k operation. This design allows each query element to attend to the most relevant neighborhood region in the keyframe feature map based on appearance similarity, rather than solely relying on spatial proximity.

In summary, standard im2col-like implementations (*e.g.*, spatial convolutions, depthwise convolutions and feature-shifting mechanisms) are data-rearrangement techniques that transform local neighborhoods into explicit, redundant patches to enable dense matrix multiplication. This process significantly increases memory consumption, as the footprint is proportional to the kernel or window size. In contrast, our direction-based unfolding treats local sampling as a logical offset mapping. By directly computing correlations between query features and sampled feature maps (or via index-based sampling), we eliminate the requirement for an intermediate "flattened" buffer. This results in a more memory-efficient and computationally streamlined implementation that preserves the inductive bias of local sampling without the overhead of physical data duplication.

3.2.4. Aggregation

Similar to (Oh et al., 2019), we retrieve corresponding features $\boldsymbol{F}$ by aggregating sampled value features $\boldsymbol{V}$ using the affinity matrix $\boldsymbol{A}$ representing a readout operation that is controlled by the sampled key $\boldsymbol{K}$ and the query $\boldsymbol{Q}$,

$$\boldsymbol{F} = \boldsymbol{V}\boldsymbol{A}(\boldsymbol{K}, \boldsymbol{Q}). \tag{10}$$

The retrieving operation maps every query element to a distribution over all N memory elements and correspondingly aggregates their values $\boldsymbol{V}$. The feature $\boldsymbol{F}$ represents the information stored in the memory is aggregated via the matching correlation score $\boldsymbol{C}$. In the following, $\boldsymbol{F}$ is fed into the decoder to generate the mask, as illustrated in Fig. 3. At the next frame, the current frame with its predicted mask can be further added into the memory as new reference, and the query $\boldsymbol{Q}$ is reused as the key feature.

3.3. Object-aware contrastive learning

In the general case, the existing matching-based approaches (Cheng & Schwing, 2022; Cheng et al., 2021b; Oh et al., 2019) mainly focus on feature matching based on a generic feature embedding ignoring the spatio-temporal consistency of the feature embedding in the training phase. In particular, our local matching strategy needs to be equipped with spatio-temporal constraints to better shape the structure of feature embedding within and across frames. Therefore, a supervised contrastive learning strategy is introduced to directly improve the consistency of query and key features for the feature-level matching and aggregation, rather than implicitly via the gradient of the decoder.

As normal, the query encoder is first adopted to map the current frame $\boldsymbol{I}_t$ to a 3D feature tensor, *i.e.*, the query $\boldsymbol{Q}$. We then computed the object-aware contrastive loss over a non-linear projection using a light weight CNN $f_{proj} : \mathbb{R}^{HW \times D} \mapsto \mathbb{R}^{HW \times C}$ such that $\boldsymbol{Z}_q = f_{proj}(\boldsymbol{Q})$, as is common in contrastive learning over convolutional feature maps (Chen et al., 2020). The projected feature $\boldsymbol{Z}_q \in \mathbb{R}^{HW \times C}$ is usually called the anchor. Likewise, the key $\boldsymbol{K}$ is also projected across the index of keyframes to obtain the projected feature $\boldsymbol{Z}_k \in \mathbb{R}^{KHW \times C}$ through the non-linear projection head f_{proj}. In this work, to identify objects in each frame, we use labels downsampled to the spatial dimensions of the feature space, denoted by $\tilde{\boldsymbol{M}}_{t'}^{k'}$ ($t' \in \{1, \ldots, K, t\}$ and $k' \in \{1, \ldots, K'\}$), where t' indexes the frames (including the current frame t and the keyframes $1, \ldots, K$) and k' indexes the objects, with K' objects in the video $\mathcal{I}$. Features projected by f_{proj}, in combination with $\tilde{\boldsymbol{M}}_{t'}^{k'}$, form the object-specific sets $\Phi_{k'}$ for each object k'.

3.3.1. Anchor-based feature set sampling

We sample a fixed number of features for each object k' present in the anchor $\boldsymbol{Z}_q$, while simultaneously collecting a positive set $\mathcal{P}_{k'}$ and a negative set $\mathcal{N}_{k'}$ for each object from its instances across K keyframes, *i.e.*, from the projected features $\boldsymbol{Z}_k$. This sampling process can be described as

$$\Gamma \sim \{i \in \cup_{k'=1}^{K'} \Phi_{k'}\}, \tag{11}$$

where a fixed number of features are selected on-the-fly rather than specified as a hyperparameter, determined by the number of feature samples from the object that contains the fewest features across the video. This is motivated by the observation that small objects or regions occupy only limited spatial positions in the feature space, yet play a significant role in feature representation learning. This heuristic ensures a balanced contribution of all objects to the loss and removes the need for hyperparameter tuning. Moreover, it reduces the computational cost of each contrastive term, enabling the cross-frame contrastive learning described next.

3.3.2. Spatio-temporal contrastive loss

In self-supervised learning, the InfoNCE (van den Oord et al., 2018) loss is computed over a set of feature vectors. Similarly, we construct positive and negative sets, denoted by $\mathcal{P}_{k'}$ and $\mathcal{N}_{k'}$ ($k' \in \{1, \dots K'\}$), from a sampled anchor-based feature set as described above. In the supervised setting we consider, these sets are determined according to the object labels $\tilde{\boldsymbol{M}}_{t'}^{k'}$. Thus, without the need for dataset-specific semantic constraints, we exploit the natural occurrences of same or different object pixels across frames in the video. The cross-frames loss for the video $\mathcal{I}$ is given by

$$\mathcal{L}_c = \frac{1}{|\Gamma|} \sum_{i \in \Gamma} \frac{1}{|\mathcal{P}_i|} \sum_{j^+ \in \mathcal{P}_i} \mathcal{L}(\boldsymbol{z}_i, \boldsymbol{z}_{j^+}), \tag{12}$$

where i is determined by Eq. (11). And the contrastive distance for each object is defined as

$$\mathcal{L}(\boldsymbol{z}_i, \boldsymbol{z}_{j^+}) = -\log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j^+}/\tau)}{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j^+}/\tau) + \sum_{j^- \in \mathcal{N}_i} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j^-}/\tau)}, \tag{13}$$

where τ is the temperature. The choice of the positive set $\mathcal{P}$ and negative set $\mathcal{N}$ varies according to the value of i. For example, for object k', the features include in $\Phi_{k'}$ form its positive set, while the remaining features in Γ form its negative set.

4. Experiments

4.1. Experimental setup

4.1.1. Datasets

We collected 42 X-ray angiography video sequences: 20 from the CADICA dataset (Jiménez-Partinen et al., 2024) and 22 from cardiac resynchronization therapy procedures performed at two hospitals. These videos capture the injection of the contrast agent and its flow through the coronary arteries or the coronary sinus along the surface of the heart. Based on these data, we established the MOSXAV dataset, where each video contains 33∼70 frames at a resolution of 512×512. Experienced radiologists annotated the vascular regions, focusing on one or two frames in which the contrast agent is most prominent for each video. The training and validation sets comprise 50 sequences (2,335 frames) with dense annotations every five frames, while the test set consists of 12 sequences (488 frames) with annotations for all frames.

In addition, we conducted comprehensive experiments on two publicly available benchmark datasets: XACV (Wu et al., 2025) and CADICA (Jiménez-Partinen et al., 2024). The XACV dataset includes 111 complete coronary artery X-ray video records from 59 patients, covering the injection, propagation through the cardiac vessels, and dissipation of the contrast agent. Each video contains an average of 86 high-resolution (512×512) frames, with an equal distribution between the left and right coronary arteries. XACV provides annotations for both blood vessels and injection catheters; however, in our method these annotations are used solely for evaluation, not for training. The CADICA dataset comprises annotated invasive coronary angiography videos from 42 patients but does not provide ground-truth segmentations. To enable quantitative evaluation, we therefore selected several videos that capture the complete dissipation of the contrast agent in the cardiac vessels and contain multiple objects, such as catheters, balloons, and surgical devices, and manually annotated them to establish segmentation ground truth for use in the subsequent experiments.

4.1.2. Evaluation metric

Following common practice (Perazzi et al., 2016; Pont-Tuset et al., 2017) in video object segmentation, we adopt the standard metrics for the X-ray angiography video segmentation: region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$ and their average $\mathcal{J}\&\mathcal{F}$. To better evaluate the generalization performance of our method on the `test` set of the MOSXAV, we also report the $\mathcal{J}$ and $\mathcal{F}$ scores separately for "seen" and "unseen" categories, denoted by subscripts s and u, respectively.

4.1.3. Implementation details

For the network architecture, we adopt ResNets (He et al., 2016) as the feature extractor, removing both the classification head and the final convolutional stage, which results in features with a stride of 16. The query encoder is based on ResNet-50, while the value encoder is based on ResNet-18. During training, we first pretrain on RGB video segmentation datasets (Perazzi et al., 2016; Pont-Tuset et al., 2017; Xu et al., 2018), and subsequently perform the main training on MOSXAV. The main training process takes approximately 21 h on two NVIDIA RTX™ 6000 GPUs. For the training loss, we use bootstrapped cross-entropy (Cheng et al., 2021a) for segmentation and the proposed spatio-temporal contrastive loss (Eq. (12)) for feature representation learning, with a loss weight ratio of 1:0.01. For optimization, we use Adam with a learning rate of 1×10^{-5} and a weight decay of 0.05. Pretraining is performed for 150K iterations with a batch size of 16, followed by 150 iterations with a batch size of 16 for the main training. For data augmentation, we apply PyTorch's random horizontal flip, random resized crop with a crop size of 384, a scale range of (0.36, 1.0), and an aspect ratio range of (0.75, 1.25), as well as color jittering with brightness, contrast, saturation, and hue factors of 0.1, 0.03, 0.03, and 0, respectively. Additionally, we apply random affine transformations with rotation between [-15, 15] degrees and shearing between [-10, 10] degrees. During inference, we use a local sampling window size of $k = 15$ for all datasets. The entire model is implemented in PyTorch (Paszke et al., 2019).

4.2. Main results

To comprehensively evaluate the superiority of our proposed method, we compare it with state-of-the-art few-shot video segmentation methods on the CADICA and XACV datasets, as well as on the `val` and `test` sets of MOSXAV (Table 1). Our method is benchmarked against eight publicly available approaches from the computer vision community. For methods originally designed for single-object segmentation, including OSVOS (Caelles et al., 2017) and Matcher (Liu et al., 2024), we extend them to the multi-object setting by splitting multiple objects into separate videos, each containing a single object.

4.2.1. CADICA dataset

We compare our method with top-performing FSVOS approaches on the public CADICA dataset using our provided segmentation ground truth. The detailed results are presented in the CADICA column of Table 1. Our method achieves substantial performance gains over existing video segmentation methods, notably outperforming the baseline FSVOS models, *i.e.*, RMem, STCN, and XMem by $+\mathbf{4.1\%}$, $+\mathbf{3.6\%}$, and $+\mathbf{3.1\%}$ in terms of $\mathcal{J}\&\mathcal{F}$, respectively. SAM-based methods such as PerSAM and Matcher perform significantly worse, primarily due to their lack of adaptation to the X-ray image domain. By contrast, the Eiseg-series methods, *i.e.*, Eiseg-EdgeFlow and Eiseg-ChestXray, exhibit better generalization to the X-ray image domain, since these matching-based approaches establish pixel correspondences based on pixel-level similarity, in a manner similar to STCN.

Table 1
Quantitative results on the CADICA, XACV and the `val` and `test` sets of the MOSXAV. The best performance scores are highlighted in bold, while gray indicates models that were not trained on X-ray video datasets.

Methods	CADICA			XACV			MOSXAV							
							`val`			`test`				
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}_s\uparrow$	$\mathcal{F}_s\uparrow$	$\mathcal{J}_u\uparrow$	$\mathcal{F}_u\uparrow$
Eiseg-EdgeFlow [ICCVW'2021] (Hao et al., 2021)	75.0	65.8	84.3	78.5	67.8	89.2	74.4	66.0	82.7	57.8	51.7	72.3	44.1	63.2
Eiseg-ChestXray [arXiv'22] (Hao et al., 2022)	73.7	64.3	83.2	76.1	65.4	86.8	73.6	65.1	82.2	54.1	49.6	68.5	40.9	57.4
PerSAM*† [arXiv'23] (Zhang et al., 2023)	21.1	17.6	24.5	25.8	21.3	30.3	20.1	16.3	24.0	6.5	3.2	4.1	8.5	10.2
Matcher*† [ICLR'24] (Liu et al., 2024)	36.0	27.3	44.7	40.2	30.5	49.9	32.5	23.9	41.1	30.4	26.2	43.5	15.2	36.7
OSVOS [CVPR'17] (Caelles et al., 2017)	72.3	59.7	84.9	74.3	64.6	84.0	71.1	60.2	82.1	47.7	37.0	62.2	37.2	54.4
STM [ICCV'19] (Oh et al., 2019)	79.5	70.4	88.6	85.3	79.8	90.7	79.7	70.7	87.6	73.1	60.9	82.1	64.1	85.0
STCN [NeurIPS'21] (Cheng et al., 2021b)	81.4	72.3	90.4	86.9	81.2	92.7	81.1	72.8	89.4	73.5	61.2	82.7	65.0	85.7
XMem [ECCV'22] (Cheng & Schwing, 2022)	81.9	72.9	91.0	88.3	82.8	93.8	81.5	73.1	89.8	74.1	63.0	83.4	64.2	84.4
PerSAM-F† [arXiv'23] (Zhang et al., 2023)	45.0	35.2	54.8	48.2	37.9	58.5	39.7	29.4	50.0	48.1	25.3	39.3	57.9	69.9
RMem [CVPR'24] (Zhou et al., 2024)	80.9	71.5	90.3	85.4	79.1	91.6	79.5	70.8	88.2	73.3	61.0	82.1	64.5	85.5
LiVOS [CVPR'25] (Liu et al., 2025)	80.7	71.5	89.8	86.2	81.4	90.9	80.4	72.0	88.7	74.0	61.0	82.2	65.9	86.7
Ours	**85.0**	**75.3**	**94.7**	**89.1**	**83.7**	**94.4**	**83.5**	**74.6**	**92.3**	**76.8**	**64.8**	**85.5**	**67.7**	**88.4**

$*$ indicates the training-free method. † indicates the method using SAM.
OSVOS and PerSAM-F are evaluated on the `test` set after online-training on the `test` set.

Table 2
Ablation study of training objective on the MOSXAV dataset, measured by mean $\mathcal{J}$.

$\mathcal{L}_{ce}$	$\mathcal{L}_c$	Mean $\mathcal{J}\uparrow$
✓		73.8
✓	✓	74.6 (↑0.8)

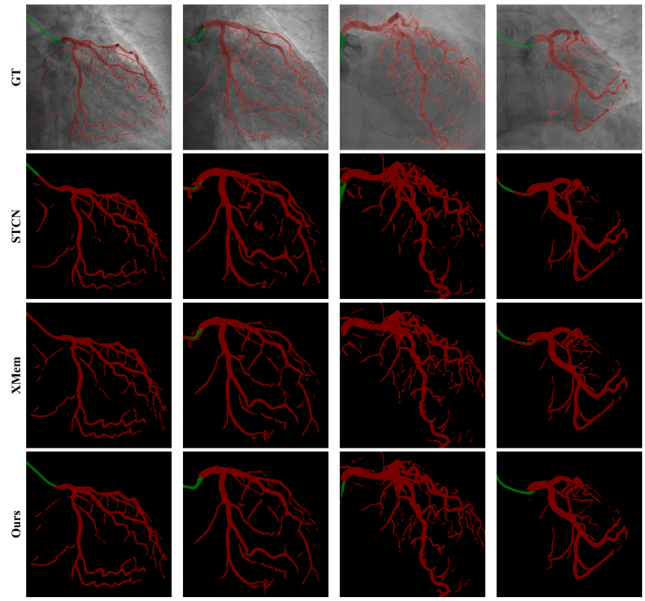

Fig. 5. Qualitative results on four challenging sequences from the XACV dataset. STCN and XMem, which rely on global matching strategies, struggle with objects represented by only a small number of pixels due to the imbalanced pixel distribution problem, whereas our method performs well on these sequences. The first row presents the ground-truth masks for different objects.

4.2.2. XACV dataset

We also report performance on the high-resolution XACV dataset in Table 1. Unlike the CADICA and MOSXAV datasets, segmentation results on XACV are evaluated without any additional training. Nevertheless, our method outperforms other FSVOS approaches, *i.e.*, LiVOS, STCN, and XMem, by **2.9%**, **2.2%**, and **0.8%**, respectively, in terms of $\mathcal{J}\&\mathcal{F}$. These results highlight two key observations: (1) the XACV dataset provides annotations for two frames in which the contrast agent is most prominent, allowing the support frame to capture the complete coronary artery structure and thereby facilitating accurate segmentation of subsequent frames; and (2) our MOSXAV dataset offers high resolution, fine-grained annotations and sufficient content diversity, covering multiple angiographic instances, which makes it a reliable resource for training X-ray image segmentation models.

4.2.3. MOSXAV val and test sets

In the MOSXAV dataset, all methods are evaluated under the same conditions, where the support frame is selected from the initial injection phase of the contrast agent and does not necessarily contain the complete coronary artery structure. This scenario is more challenging and more closely reflects real-world clinical practice. Importantly, the `test` set of MOSXAV contains unseen object classes in the `train` set, allowing for evaluation of the model's generalizability.

As shown in Table 1 (MOSXAV column), our method achieves the best performance among all video segmentation approaches on this challenging multi-object dataset. On the `val` set, our method attains an $\mathcal{J}\&\mathcal{F}$ score of **83.5%**, outperforming the second-best method (XMem) and the third-best method (STCN) by **2.0%** and **2.4%**, respectively. On the `test` set, particularly for unseen object classes, our approach demonstrates consistent performance gains over XMem and LiVOS, improving the $\mathcal{J}\&\mathcal{F}$ score from 74.1% to 76.8% and from 74.0% to 76.8%, respectively.

4.2.4. Qualitative results

Fig. 5 provides a qualitative comparison of our method against STCN and XMem on representative examples from the XACV dataset. We observe that our approach effectively handles diverse and challenging scenarios, producing more accurate results. Fig. 6 presents a qualitative comparison on the `val` and `test` sets of MOSXAV. Compared with state-of-the-art FSVOS methods, our approach achieves more stable and accurate mask-tracking results, even in challenging scenarios involving overlapping vessels (*v01*), severe cardiac motion (*v19*) and unseen object classes under low-dose fluoroscopy (*v06*).

4.3. Ablation study

To demonstrate the effectiveness of the local sampling module in our method, we perform an ablation study on the `val` set of MOSXAV. The evaluation criterion is the mean region similarity ($\mathcal{J}$) and frames par second (FPS).

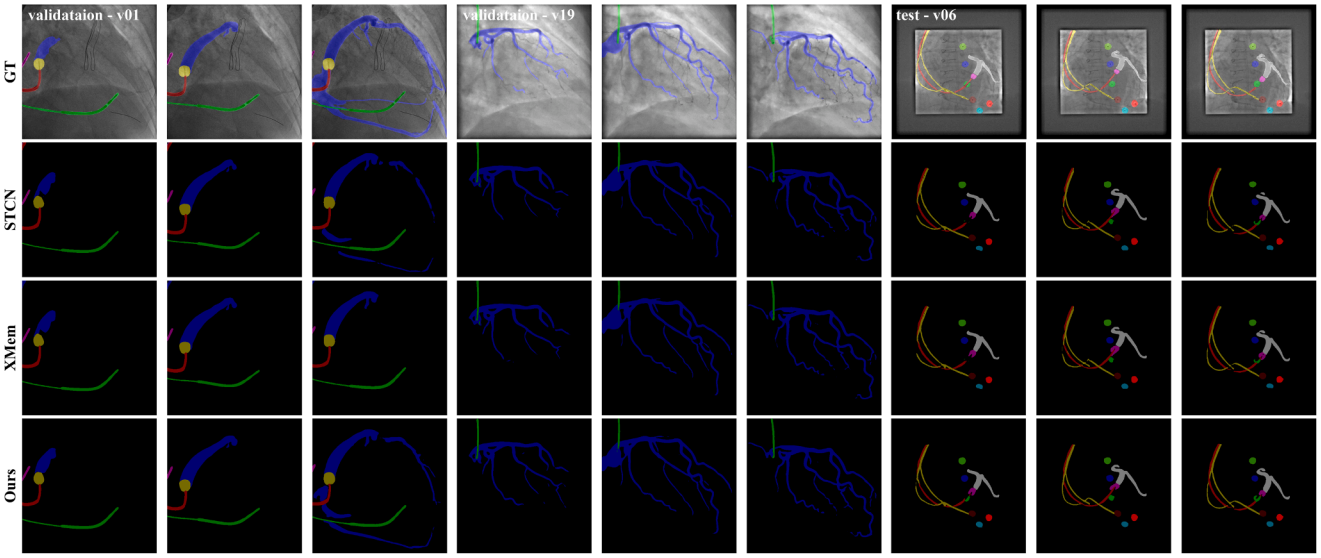

Fig. 6. Qualitative results on three challenging sequences from the `val` and `test` sets of MOSXAV. The two validation sequences (*i.e.*, *v01* and *v19*) present complex scenarios with overlapping vessels and severe cardiac motion. The `test` sequence (*i.e.*, *v06*) contains unseen object classes not present in the training set, specifically six spherical objects, and was captured using low-dose fluoroscopy. The first row shows the ground truth masks for the different objects.

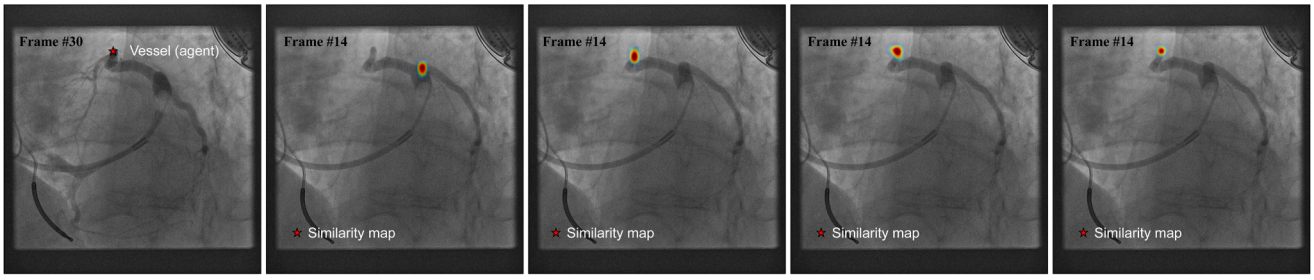

Fig. 7. Visualization of the pixels in the keyframe with the highest similarity scores relative to the selected coronary vessel pixel (red star - ★) in the query frame during training. Based on the similarity scores between the selected pixel in the query frame and the most relevant pixels in keyframe, the model gradually learns to focus on the vessel structure while suppressing responses to irrelevant distractors.

Table 3
Ablation studies on the MOSXAV, measured by mean $\mathcal{J}$ and FPS.

(a) Sampling operation function

Sampling Strategies	Mean $\mathcal{J}$ ↑	FPS ↑		
		RTX™ 6000 GPU	i9-13900H CPU	R9 7940HS CPU
baseline	72.8	20	0.289±0.006	0.283±0.004
feature shift	72.4 (↓0.4)	10 (↓10)	0.268±0.003 (↓0.021)	0.257±0.002 (↓0.026)
depth-wise conv	73.3 (↑0.5)	19 (↓1)	0.282±0.001 (↓0.007)	0.275±0.005 (↓0.008)
deformable conv	73.8 (↑1.0)	16 (↓4)	-	-
2D neighborhood attention	74.5 (↑1.7)	22 (↑2)	-	-
unfold (**ours**)	**74.6** (↑1.8)	**25** (↑5)	0.301±0.004 (↑0.012)	0.293±0.001 (↑0.010)

(b) Neighborhood region

Sampling Size k	Mean $\mathcal{J}$ ↑	FPS ↑
k=5	73.1	**43**
k=7	73.3	37
k=9	74.5	32
k=13	74.5	27
k=15	**74.6**	25
k=17	74.5	22
k=19	74.3	19

(c) Keyframe sampling interval

r-frames	Mean $\mathcal{J}$ ↑	FPS ↑
r=1	74.2	10
r=2	74.7	16
r=3	74.2	20
r=5	73.7	23
r=6	**74.6**	25
r=8	74.5	27
r=10	73.6	**28**

- denotes that the operation is exclusively optimized for GPU via specialized CUDA kernels and lacks a corresponding CPU implementation.

4.3.1. Training objective

We investigate our overall training objective, as described in Section 4.1.3, which consists of the cross-entropy loss $\mathcal{L}_{ce}$ and the spatio-temporal contrastive loss $\mathcal{L}_c$. As shown in Table 2, the model trained with $\mathcal{L}_{ce}$ alone achieves a mean $\mathcal{J}$ score of 73.8%. Incorporating $\mathcal{L}_c$ yields an additional improvement of 0.8%, highlighting the benefit of explicitly shaping the feature representation.

4.3.2. Sampling strategies

To evaluate the efficacy of the local sampling module $\phi(\cdot;\cdot)$ in Eq. (7), we systematically replace it with alternative operations, specifically the feature shifting mechanism (*i.e.*, spatial feature slicing) utilized in Window Attention (Liu et al., 2021) and the depth-wise convolution adopted in Slide Attention (Pan et al., 2023). For a fair comparison, these alternatives are implemented using the official PyTorch implementations from their respective open-source repositories. We adopt the global feature matching strategy of STCN (Cheng et al., 2021b) as our baseline. Furthermore, we implement the local sampling process using specialized CUDA kernels (*e.g.*, deformable convolution (Dai et al., 2017) and neighborhood attention (Hassani et al., 2023)) and benchmark their performance on an RTX™ 6000 GPU. Additionally, we assess the CPU inference efficiency of the local sampling module using Intel® Core™ i9-13900H and AMD Ryzen™ 9 7940HS processors. Results are summarized in Table 3(a). For CPU inference speed (FPS), each implementation is executed ten times under identical hyperparameters settings, awith the mean and standard deviation reported.

Compared with the baseline, our local sampling module implemented via the unfold sampling function (Eq. (7)) improves segmentation performance by **1.8%** in terms of mean $\mathcal{J}$. More importantly, it increases inference speed to **25** FPS on the GPU (*vs.* 20 FPS for the baseline) and to **0.301** FPS and **0.293** FPS on the Intel® Core™ i9-13900H CPU and the AMD Ryzen™ 9 7940HS CPU, respectively (*vs.* 0.289 and 0.283 for the baseline). To improve efficiency, we first partition the image into patches, similar to the patch embedding module in ViT (Dosovitskiy et al., 2021), and then compute similarity within each local window. The main motivation for this design is that applying im2col-like local sampling individually to every pixel is prohibitively expensive. Regarding clinical utility, our current PyTorch implementation requires approximately 3.32 and 3.41 s per image on the aforementioned Intel and AMD CPUs. While this execution speed is not yet optimized for real-time intraoperative use, it is well-suited for offline analysis and automated batch processing within the research phase. Our method achieves a performance gain over the global matching baseline while maintaining competitive accuracy, suggesting it can be effectively integrated into clinical workflows that prioritize reliability over latency, such as retrospective cohort studies or pre-procedural planning. Consistent with this observation, we found that the feature-shift operation based on im2col-like functions yield both lower segmentation accuracy and higher computational costs. For fixed and deformable convolution-based sampling, the learnable bias terms fail to capture effective inductive biases due to the depth-wise nature of these operations. In our implementation, we fix the kernel weights to 1 and optimize only the bias parameters. Compared with the baseline, these two sampling operations improve accuracy by 0.5% and 1.0%, respectively, both resulted in reduced inference speeds across GPU and CPU architectures. In contrast, while 2D neighborhood attention achieves strong performance and efficiency, its deployment, much like deformable convolution, remains constrained by a dependency on device-specific CUDA implementations.

Fig. 7 visualizes the evolution of the learned correspondences process, which is crucial for understanding how feature representations develop during training. In the early iterations, the model fails to capture the relevant regions due to the presence of visually similar distractors and textured backgrounds (*e.g.*, the similarity map at 100 iterations). As training progresses, the most relevant pixels gradually concentrate on the vessel structures, while responses to irrelevant distractors are increasingly suppressed. These qualitative results demonstrate that our local sampling strategy effectively captures meaningful contextual information. Notably, even at 10K iterations, the model is able to identify most relevant of the vessel structure, indicating that the proposed local sampling mechanism enables efficient learning of discriminative features.

4.3.3. Sampling size

Table 3(b) reports the performance of our approach with respect to the sampling region size k. As k increases, the mean $\mathcal{J}$ first improves and then declines. Notably, using a larger region ($k=9$) yields a clear performance gain (73.1%→74.5%). The score further improves at $k=15$; however, increasing k beyond 15 results in a slight drop in performance. Based on this observation, we empirically set $k=15$, which provides the best trade-off between accuracy and computational cost. Smaller sampling sizes achieve higher FPS, as the matrix multiplication on $\mathcal{M}=\{(i,j),c_{ij}\}$ in Section 3.2.1 involves fewer operations.

4.3.4. Keyframes sampling

We compare different keyframe sampling intervals r in Table 3(c). For $r=1$, every frame in the video is selected as a support frame and stored in the external memory to build the feature memory, yielding a baseline score of 74.2%. As r increases, the FPS improves due to the reduced size of the correspondence matrix $\boldsymbol{C}$ in Section 3.2.1, while the mean $\mathcal{J}$ initially rises and then drops. Based on this observation, we empirically set $r=6$ to achieve a better trade-off between accuracy and computational cost.

5. Conclusion

In this paper, we propose an FSVOS method for angiography video segmentation, aiming to minimize the prohibitive costs of expert annotation. By focusing on localized neighborhood regions, our method achieves superior efficiency and generalizability in feature matching. To ensure broad applicability across diverse hospital computing infrastructures, we replaced inefficient standard im2col-like implementations and hardware-dependent kernels with a non-parametric local sampling operation. Extensive evaluations on the CADICA, XACV, and MOSXAV datasets demonstrate that our approach outperforms state-of-the-art video segmentation methods, providing the precise vessel delineation required for clinical decision-making.

Despite these methodological advances, translating a retrospective research prototype into a real-world clinical application presents significant challenges, particularly regarding data privacy, system interoperability, regulatory compliance, and real-time deployment. To bridge this gap, our future work will focus on three strategic fronts: (1) Clinical prototyping: We aim to develop a DICOM-compatible plugin via open-source platforms such as 3D Slicer, facilitating intuitive clinician interaction and seamless data integration. (2) Regulatory and security standards: We will further optimize inference latency and strengthen data security protocols to align with stringent medical device regulatory standards. (3) High-performance deployment: To achieve real-time capability, we will transition from our current Python-based implementation to high-efficiency C++ frameworks, specifically leveraging LibTorch and ONNX Runtime to enable advanced parallel processing and hardware acceleration. We believe this work establishes a robust foundation for "human-in-the-loop" AI systems, ultimately enhancing standard diagnostic workflows with reliable, high-performance assistance.

CRediT authorship contribution statement

Lin Xi: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation; **Yingliang Ma:** Supervision, Project administration, Data curation; **Xiahai Zhuang:** Writing – review & editing, Supervision, Methodology.

Declaration of competing interests

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.

Acknowledgement

This work was supported by EPSRC UK (EP/X023826/1).

References

Ambrosini, P. et al. (2017). Fully automatic and real-time catheter segmentation in x-ray fluoroscopy. In *Medical image computing and computer-assisted intervention (MICCAI)* (pp. 577–585).

Baur, C. et al. (2016). CathNets: Detection and single-view depth prediction of catheter electrodes. In *Medical imaging and augmented reality* (pp. 38–49).

Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., & Van Gool, L. (2017). One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5320–5329).

Chellapilla, K., Puri, S., & Simard, P. (2006). High performance convolutional neural networks for document processing. In *Tenth international workshop on frontiers in handwriting recognition*.

Chen, C., Panda, R., & Fan, Q. (2022). RegionViT: Regional-to-local attention for vision transformers. In *International conference on learning representations (ICLR)*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. E. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the international conference on machine learning (ICML)* (pp. 1597–1607). (*vol. 119*).

Cheng, H. K., & Schwing, A. G. (2022). XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 640–658). (*vol. 13688*).

Cheng, H. K., Tai, Y.-W., & Tang, C.-K. (2021a). Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5555–5564).

Cheng, H. K., Tai, Y.-W., & Tang, C.-K. (2021b). Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Advances in neural information processing systems (neurIPS)* (pp. 11781–11794).

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., & Shelhamer, E. (2014). cuDNN: Efficient primitives for deep learning. arXiv:1410.0759.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 764–773).

van den, O. A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations (ICLR)*.

Gu, Z., Cheng, J., Fu, H., Zhou, K., Hao, H., Zhao, Y., Zhang, T., Gao, S., & Liu, J. (2019). CE-Net: Context encoder network for 2D medical image segmentation. *IEEE Transactions on Medical Imaging*, *38*(10), 2281–2292.

Gutmann, M., & Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics (AISTATS)* (pp. 297–304). (*vol. 9*).

Hao, D., Ding, S., Qiu, L., Lv, Y., Fei, B., Zhu, Y., & Qin, B. (2020). Sequential vessel segmentation via deep channel attention network. *Neural Networks*, *128*, 172–187.

Hao, Y., Liu, Y., Chen, Y., Han, L., Peng, J., Tang, S., Chen, G., Wu, Z., Chen, Z., & Lai, B. (2022). EISeg: An efficient interactive segmentation tool based on paddlepaddle. arXiv preprint arXiv:2210.08788.

Hao, Y., Liu, Y., Wu, Z., Han, L., Chen, Y., Chen, G., Chu, L., Tang, S., Yu, Z., Chen, Z., & Lai, B. (2021). EdgeFlow: Achieving practical interactive segmentation with edge-guided flow. In *Proceedings of the IEEE international conference on computer vision workshops (ICCVW)* (pp. 1551–1560).

Hassani, A., Walton, S., Li, J., Li, S., & Shi, H. (2023). Neighborhood attention transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 6185–6194).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 770–778).

Hénaff, O. J. (2020). Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the international conference on machine learning (ICML)* (pp. 4182–4192). (*vol. 119*).

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., & Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *International conference on learning representations (ICLR)*.

Hu, Y.-T., Huang, J.-B., & Schwing, A. (2017). MaskRNN: Instance level video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, *30*.

Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on theory of computing* (pp. 604–613).

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). CAFFE: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM international conference on multimedia (ACM MM)* (pp. 675–678).

Jiménez-Partinen, A., Molina-Cabello, M. A., Thurnhofer-Hemsi, K., Palomo, E. J., Rodríguez-Capitán, J., Molina-Ramos, A. I., & Jiménez-Navarro, M. (2024). CADICA: A new dataset for coronary artery disease detection by using invasive coronary angiography. *Expert Systems*, *41*(12), e13708.

Jun, T. J., Kweon, J., Kim, Y.-H., & Kim, D. (2020). T-Net: Nested encoder-decoder architecture for the main vessel segmentation in coronary angiography. *Neural Networks*, *128*, 216–233.

Khan, T. M., Naqvi, S. S., Robles-Kelly, A., & Razzak, I. (2023). Retinal vessel segmentation via a multi-resolution contextual network and adversarial learning. *Neural Networks*, *165*, 310–320.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. In *Advances in neural information processing systems (neurIPS)* (pp. 18661–18673). (*vol. 33*).

Kim, T. et al. (2022). A learning-based, region of interest-tracking algorithm for catheter detection in echocardiography. *Computerized Medical Imaging and Graphics*, *100*, 102106.

Liu, Q., Wang, J., Yang, Z., Li, L., Lin, K., Niethammer, M., & Wang, L. (2025). LiVOS: Light video object segmentation with gated linear matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 8668–8678).

Liu, Y., Zhu, M., Li, H., Chen, H., Wang, X., & Shen, C. (2024). Matcher: Segment anything with one shot using all-purpose feature matching. In *International conference on learning representations (ICLR)*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 9992–10002).

Lu, X., Wang, W., Ma, C., Shen, J., Shao, L., & Porikli, F. (2019). See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 3618–3627).

Ma, Y., Wang, H., Shen, H., Duan, S., & Wen, S. (2025). Analog spiking U-Net integrating CBAM&vit for medical image segmentation. *Neural Networks*, *181*, 106765.

Maninis, K.-K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., & Van Gool, L. (2019). Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(6), 1515–1530.

Mnih, A., & Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems (neurIPS)* (pp. 2265–2273). (*vol. 26*).

Nguyen, A. et al. (2020). End-to-end real-time catheter segmentation with optical flow-guided warping during endovascular intervention. In *IEEE international conference on robotics and automation (ICRA)* (pp. 9967–9973).

Oh, S. W., Lee, J.-Y., Xu, N., & Kim, S. J. (2019). Video object segmentation using space-time memory networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 9225–9234).

Pan, X., Ye, T., Xia, Z., Song, S., & Huang, G. (2023). Slide-transformer: Hierarchical vision transformer with local self-attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 2082–2091).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems (neurIPS)* (pp. 8024–8035). (*vol. 32*).

Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., & Sorkine-Hornung, A. (2017). Learning video object segmentation from static images. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3491–3500).

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., & Sorkine-Hornung, A. (2016). A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 724–732).

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., & Van Gool, L. (2017). The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., & Shlens, J. (2019). Stand-alone self-attention in vision models. In *Advances in neural information processing systems (neurIPS)*. (*vol. 32*).

Ranne, A. et al. (2024). AiAReSeg: Catheter detection and segmentation in interventional ultrasound using transformers. In *IEEE international conference on robotics and automation (icra)* (pp. 8187–8194).

Robinson, A., Lawin, F. J., Danelljan, M., Khan, F. S., & Felsberg, M. (2020). Learning fast and robust target models for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 7404–7413).

Roh, B., Shin, J., Shin, W., & Kim, S. (2022). Sparse DETR: Efficient end-to-end object detection with learnable sparsity. In *International conference on learning representations (ICLR)*.

Ronneberger, O. et al. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention (MICCAI)* (pp. 234–241).

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., & Brain, G. (2018). Time-contrastive networks: Self-supervised learning from video. In *International conference on robotics and automation (ICRA)* (pp. 1134–1141).

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems (neurIPS)* (pp. 1849–1857). (*vol. 29*).

Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive multiview coding. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 776–794). (*vol. 12356*).

Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., & Lucic, M. (2020). On mutual information maximization for representation learning. In *International conference on learning representations (ICLR)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems (neurIPS)* (pp. 5998–6008). (*vol. 30*).

Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., & Giro-i Nieto, X. (2019). RVOS: End-to-end recurrent network for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5272–5281).

Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., & Chen, L.-C. (2019). FEELVOS: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 9473–9482).

Wang, Q., Zhang, L., Bertinetto, L., Hu, W., & Torr, P. H. S. (2019). Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1328–1338).

Wang, T., Yuan, L., Chen, Y., Feng, J., & Yan, S. (2021). PNP-DETR: Towards efficient visual analysis with transformers. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 4641–4650).

Wu, C.-H., Chen, S.-H., Hu, C.-Y., Wu, H.-Y., Chen, K.-H., Chen, Y.-Y., Su, C.-H., Lee, C.-K., & Liu, Y.-L. (2025). DeNVer: Deformable neural vessel representations for unsupervised video vessel segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 15682–15692).

Wu, X. et al. (2015). Fast catheter segmentation from echocardiographic sequences based on segmentation from corresponding x-ray fluoroscopy for cardiac catheterization interventions. *IEEE Transactions on Medical Imaging*, *34*(4), 861–876.

Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3733–3742).

Xi, L., Chen, W., Wu, X., Liu, Z., & Li, Z. (2022). Implicit motion-compensated network for unsupervised video object segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(9), 6279–6292.

Xi, L., Chen, W., Wu, X., Liu, Z., & Li, Z. (2024). Online unsupervised video object segmentation via contrastive motion clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, *34*(2), 995–1006.

Xi, L., Ma, Y., Koland, E., Howell, S., Rinaldi, A., & Rhode, K. S. (2025). Catheter detection and segmentation in x-ray images via multi-task learning. *International Journal of Computer Assisted Radiology and Surgery*, *21*, 1–11.

Xie, J., Xiang, J., Chen, J., Hou, X., Zhao, X., & Shen, L. (2022). C2AM: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 979–988).

Xu, H., & Wu, Y. (2024). G2ViT: Graph neural network-guided vision transformer enhanced network for retinal vessel and coronary angiograph segmentation. *Neural Networks*, *176*, 106356.

Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., & Huang, T. (2018). YouTube-VOS: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327.

Yang, H. et al. (2019). Catheter localization in 3D ultrasound using voxel-of-interest-based convnets for cardiac intervention. *International Journal of Computer Assisted Radiology and Surgery*, *14*, 1069–1077.

Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., & Gao, J. (2021a). Focal attention for long-range interactions in vision transformers. In *Advances in neural information processing systems (neurIPS)* (pp. 30008–30022). (*vol. 34*).

Yang, Z., Wei, Y., & Yang, Y. (2020). Collaborative video object segmentation by foreground-background integration. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 332–348). (*vol. 12350*).

Yang, Z., Wei, Y., & Yang, Y. (2021b). Associating objects with transformers for video object segmentation. In *Advances in neural information processing systems (neurIPS)* (pp. 2491–2502).

Zhang, F., Torr, P. H. S., Ranftl, R., & Richter, S. R. (2021). Looking beyond single images for contrastive semantic segmentation learning. In *Advances in neural information processing systems (neurIPS)* (pp. 3285–3297).

Zhang, P., Sun, H., Zhang, Z., Cheng, X., Zhu, Y., & Zhang, J. (2025). Privacy-preserving recommendations with mixture model-based matrix factorization under local differential privacy. *IEEE Transactions on Industrial Informatics*, *21*(7), 5451–5459.

Zhang, R., Jiang, Z., Guo, Z., Yan, S., Pan, J., Ma, X., Dong, H., Gao, P., & Li, H. (2023). Personalize segment anything model with one shot. arXiv preprint arXiv:2305.03048.

Zhao, X., Vemulapalli, R., Mansfield, P. A., Gong, B., Green, B., Shapira, L., & Wu, Y. (2021). Contrastive learning for label efficient semantic segmentation. In *2021 IEEE/CVF International conference on computer vision (ICCV)* (pp. 10603–10613).

Zheng, Z., Shi, Y., Li, C., Hu, J., Zhu, X. X., & Mou, L. (2024). Reducing annotation burden: Exploiting image knowledge for few-shot medical video object segmentation via spatiotemporal consistency relearning. In *Proceedings of medical image computing and computer-assisted intervention (MICCAI)* (pp. 272 – 282).

Zhou, J., Pang, Z., & Wang, Y.-X. (2024). RMem: Restricted memory banks improve video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 18602–18611).

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable DETR: Deformable transformers for end-to-end object detection. In *International conference on learning representations (ICLR)*.