A Deep Learning-based Interactive Medical Image Segmentation Framework with Sequential Memory

Ivan Mikhailov[*,a,b], Benoit Chauveau[b,c],
Nicolas Bourdel[a,b,c], Adrien Bartoli[a,b,c]

[a] *EnCoV, Institut Pascal, Université Clermont Auvergne, Clermont-Ferrand, 63000, France*
[b] *SurgAR, 22 All. Alan Turing, Clermont-Ferrand, 63000, France*
[c] *CHU de Clermont-Ferrand, Clermont-Ferrand, 63000, France*

Abstract

Background and objective. Image segmentation is an essential component in medical image analysis. The case of 3D images such as MRI is particularly challenging and time consuming. Interactive or semi-automatic methods are thus highly desirable. However, existing methods do not exploit the typical sequentiality of real user interactions. This is due to the interaction memory used in these systems, which discards ordering. In contrast, we argue that the order of the user corrections should be used for training and lead to performance improvements.

Methods. We contribute to solving this problem by proposing a general multi-class deep learning-based interactive framework for image segmentation, which embeds a base network in a user interaction loop with a user feedback memory. We propose to model the memory explicitly as a sequence of consecutive system states, from which the features can be learned, generally learning from the segmentation refinement process. Training is a major difficulty owing to the network's input being dependent on the previous output. We adapt the network to this loop by introducing a virtual user in the training process, modelled by dynamically simulating the iterative user feedback.

Results. We evaluated our framework against existing methods on the complex task of multi-class semantic instance female pelvis MRI segmentation

[*]Corresponding author. Tel.: +33 7 83 93 88 85
Email address: ivanmikhailov.mail@gmail.com (Ivan Mikhailov[*])

with 5 classes, including up to 27 tumour instances, using a segmentation dataset collected in our hospital, and on liver and pancreas CT segmentation, using public datasets. We conducted a user evaluation, involving both senior and junior medical personnel in matching and adjacent areas of expertise. We observed an annotation time reduction with 5'56" for our framework against 25' on average for classical tools. We systematically evaluated the influence of the number of clicks on the segmentation accuracy. A single interaction round our framework outperforms existing automatic systems with a comparable setup. We provide an ablation study and show that our framework outperforms existing interactive systems.
Conclusions. Our framework largely outperforms existing systems in accuracy, with the largest impact on the smallest, most difficult classes, and drastically reduces the average user segmentation time with fast inference at 47.2 ± 6.2 ms per image.

Keywords: Interactive segmentation, Deep Learning, MRI, CT, RNN

1. Introduction

Image segmentation is an essential component of many visual processing systems, which involves classifying each pixel or, equivalently, delineating the regions containing pixels of the same class. In medical image analysis, the images are often patient scans from modalities such as MRI (Magnetic Resonance Imaging) or CT (Computed Tomography). MRI segmentation is a tremendously difficult task, owing to it being 3D, low contrast, noisy, low resolution and artifacted. Existing segmentation approaches can be divided into three settings based on user involvement: manual, automatic and interactive. The manual approach is the most time-consuming, as each pixel has to be attributed a label independently, which may require hours for a single MRI. It is error-prone and infeasible in the clinical environment. At the other extreme lies the automatic approach, which works without user involvement. This strongly limits its applicability, as a clinician operator shall validate and possibly edit the result before its use in a therapeutic act. The interactive approach trades-off manual and automatic features: it typically involves an automatic part with an extent of user control. Both aspects are crucial for systems designed for the clinical environment, where there generally are three main constraints: (1) decision-making should be human-controlled; (2) time is limited; (3) high accuracy is desired. Creating interactive systems

addressing these three concerns is therefore essential to simplify, speed up and secure segmentation in the clinical environment.

The automatic approach is largely dominated by deep learning, which overturned classical methods over the last decade in many segmentation tasks [4, 35]. In contrast, interactive deep learning methods present specific difficulties and have yet received relatively limited attention [36]. Concretely, deep learning interactive segmentation requires embedding a network in an interactive-loop system allowing the user to interact. Indeed, the network inputs must include the user feedback, which depends on the network outputs. This creates a dependency between the inputs and outputs of the network, which is poorly resolved by a regular training process from static data. Specifically, the input configuration and training process of interactive existing deep learning methods do not reflect how the user interactions are provided at test time. They consequently do not take full advantage of having user interactions as input, missing two key aspects: (a) realistic interaction simulation - real interactions are positioned rationally, but often scarcely and randomly distributed, an aspect which is not modelled in existing simulation approaches for training; (b) temporal interaction information - inherently present at all times in the real world, but overlooked by the existing interactive segmentation methods.

Dynamics or temporal information are additional cues typically used in video segmentation and tracking methods, which take advantage of the order and similarity of adjacent video frames. In interactive segmentation, a user interacts depending on the current segmentation result they observe, which is conditioned by both their interactions and the system's result so far. Hence, the ordering of interactions is highly important and should not be altered, as they otherwise become less informative. Intuitively, capturing the interaction order should be beneficial in any interactive framework, including interactive segmentation. We propose a general deep learning interactive segmentation framework and training methods for multi-class semantic instance segmentation. Our system consists of an embedded network, a user interaction loop and an interaction memory. First, the user reviews the current segmentation result and, if satisfied, accepts. Otherwise, the user may quickly make simple corrections by placing points or strokes to refine the segmentation, which is achieved by a special input configuration of the embedded network. Indeed, this network inputs the image, user correction masks, and possibly other memorised parameters, and outputs the segmentation probability maps. The system then loops back to the user review step, whilst updating

the interaction memory to keep track of the user corrections throughout the interactions.

In practice, the additional temporal information is represented by a neural network input structured differently than existing work. Existing works store all the interactions in the same mask, discarding the order of the interactions and hence the temporal information. We call such input structures Cumulative Interaction Memory (CIM). In contrast, we propose Sequential Interaction Memory (SIM), which stores a sequence of states instead, where each state is a pair of user input and corresponding segmentation output. Simply put, SIM is a sequence of ordered user actions and their results in time and carries temporal information by definition. The proposed architecture takes an image and a SIM as inputs and produces a segmentation as output. The system then adds this segmentation along with the latest user interaction mask to the SIM and proceeds to the next interaction round. In practice, SIM is represented by a tensor of a certain size, depending on the memory size, and is used as an input to the network at all times.

Our contributions are threefold. First, we propose a general deep learning-based interactive multi-class semantic image segmentation framework with a user interaction loop. Second, we propose a sequential interaction memory, which keeps track of the segmentation results and user corrections, maintaining sequentiality within the system. Third, we propose a general dynamic data training process, which simulates the correction-focused and sequential nature of human user feedback by learning from interaction sequences of a virtual user and minimises interaction-dependence, improving performance.

We demonstrate our framework in three tasks. The first task is multi-class semantic MRI segmentation of the female pelvis, for which we created a new dataset collected in our hospital. We validate the results against automatic and existing interactive systems with the standard metrics and perform an ablation study of our system's components. We report results of a user study conducted with both senior and junior medical users in terms of both standard metrics and elapsed time, using a specifically developed graphical user interface connected to our system. The second and third tasks are respectively the multi-class semantic liver and pancreas CT segmentation, using the "Liver Tumours" and "Pancreas Tumour" medical segmentation decathlon datasets [45]. We validate the results against automatic approaches participating in the ongoing medical segmentation decathlon challenge [2]. For these tasks, we instantiate our system with an existing encoder-decoder architecture optionally featuring RNN [39] modules.

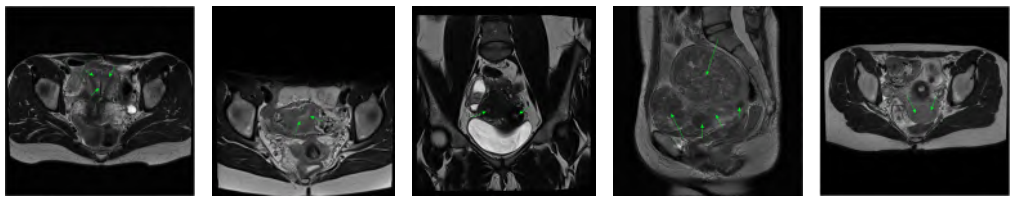

Figure 1: Female pelvis MRI dataset samples, with main difficulties indicated with green arrows, series 1 to 5: **(1)** presence of an IUD, not seen in the training set; **(2,5)** unclear contours, blurriness of the uterine cavity; **(3)** similarity of the uterine (left) and cervix cavities (right); **(4)** strong uterus deformation due to the tumours, with here five tumours.

This paper extends a shorter workshop version [33]. Specifically, we have extended it along five ways. First, we have included a third contribution, the dynamic data generation process, focused on improving generalisation and the impact of individual interactions on the performance. Second, we have extended the experimental evaluation comparison with existing approaches for all three datasets. Third, we have included a study of the influence of the number of provided user interactions on the framework's performance, including when using the framework in the automatic mode with 0 clicks provided. Fourth, we have finalised the initial preliminary user study, extending it from three to eight medical experts. Lastly, we have refined the presentation of the overall framework's architecture and its details.

2. Related Work

We review classical and deep learning approaches to medical image segmentation, distinguishing automatic and interactive approaches for each. Classical automatic segmentation encompasses a wide variety of methods [54]. Their performance is usually insufficient to achieve clinically-acceptable accuracy and they have been largely taken over by deep learning in many tasks [4]. In contrast, classical interactive methods are still widely used. The most well known ones are probably the Graph Cuts [3], Random Walker [16] and Geodesic Image Segmentation (GeoS) [10]. They achieve acceptable performance for simple cases. However, medical data often feature structures with complex shapes and poorly defined contours, noise and artefacts. This results in a substantial increase of user time required to perform segmentation and limited achievable accuracy.

Deep learning-based automatic segmentation includes a multitude of methods. A review and evaluation of over 100 methods [34] was conducted with ResNet [19] extensively used as a backbone, represented by EMANet [29]. It achieved top scores on the PASCAL VOC dataset together with [55], which adopts NAS-FPN [14] with EfficientNet-L2 [51]. Most of the models use an encoder-decoder architecture [34]. This includes the U-Net [37], with a wide spectrum of applications [43], and recent variants [13, 44] reaching top positions in the BraTS challenge 2021. Automatic MRI segmentation was attempted for various targets, including the kidney [25], the prostate [17] and brain tumours [18]. These methods demonstrate state-of-the-art performance in their respective tasks. However, they are automatic and do not allow the user to interact. Automatic segmentation is highly appropriate in applications which cannot involve user interactions in essence, such as real-time organ tracking. In contrast, many applications require validation and corrections from a certified user. For such applications, the direct use of automatic deep learning methods is inappropriate.

The integration of deep learning within interactive segmentation systems is a major challenge. A simple approach is to use a classical interactive method to post-process the result from an automatic deep learning method [49] or correct it manually [41]. Such systems inherit the intrinsic limitations of the chosen classical method. A more advanced approach is to use a neural network to process user feedback in an interactive-loop system [1, 50, 52, 53, 30, 40, 21]. These methods use a network which takes the image and user interaction masks as inputs. Training is challenging owing to the loop. Existing approaches generate user interaction masks from labelled data, either statically before training or dynamically during training, or attempt to avoid training altogether. Static data training methods [50, 52, 53] limit the system's generalisation and interaction effectiveness. Intuitively, a real user interacts based on the current segmentation they observe. In other words, the goal of the user is to improve upon what is already there. Hence, it is sound that mimicking this mechanism of acting sequentially is more faithful and true-to-practice than the previous mechanism, namely Static Data Generation (SDG), not taking past segmentations into account.

Dynamic data training methods [1, 30, 47, 21, 26] mimic this mechanism and simulate user interactions by sampling missegmented regions. This is done once from a single prediction [1] or from the latest segmentation result [30, 47]. Usually, such methods rely on a virtual user, which generates user input artificially at training time, since the involvement of real

users is not feasible. These methods diversify the training data and improve performance. However, previous works using dynamic data training have two shortcomings: first, they consider only individual classes for click placement, which is not well-adapted to the medical scan data naturally containing multi-instance or multi-component structures, and second, they do not handle multi-class multi-label multi-instance problems with multiple components per class. These problems make medical scan segmentation challenging, as they incur the fragmentation of classes into multiple components, all compounded by the inherent noise, variability and complexity of medical image scenes. In order to exemplify their terms, consider for instance, the female pelvis MRI dataset we assembled. It has multiple properties typical for medical scan datasets, namely (1) multi-class - the dataset contains multiple classes (that is, `uterus`, `bladder`, `tumour` and `cavity`); (2) multi-label - certain classes overlap (e.g. `uterus` contains `tumour` and `cavity`); (3) multi-instance - certain classes contain multiple instances (there can be multiple tumours per image); (4) multi-component - an instance of each class in the image might be split into multiple closed contours due to medical scan slicing and the shape of the object in question.

Alternatively, training-less methods were proposed to bypass the training challenges [22, 46]. Specifically, they use an automatic segmentation network interactively via inference-time optimisation and improve performance. However, these methods have certain drawbacks. First, they require backward passes using gradients, leading to a computational overhead. Second, their applicability is limited because widely used frameworks often lack support for the backward passes on mobile devices. These two factors make it difficult to apply them in practice, provided the limited availability of the high-performance GPUs in clinical workstations and laptops. An open-source interactive segmentation platform [12] was recently made available, which offers both deep learning-based [49, 40] and classical methods [3], inheriting their limitations.

The existing methods do not reproduce the typical sequentiality of real user interactions. The lack of sequentiality is a consequence of the interaction memory used in these systems, which simply accumulates the user corrections, discarding ordering. In contrast, we argue that the order of the user corrections can be directly used for training and lead to performance improvements. In short, the rationale is that the order in which the user corrects the segmentation in an interactive system depends on the current segmentation estimate. The order of interactions can thus not be changed

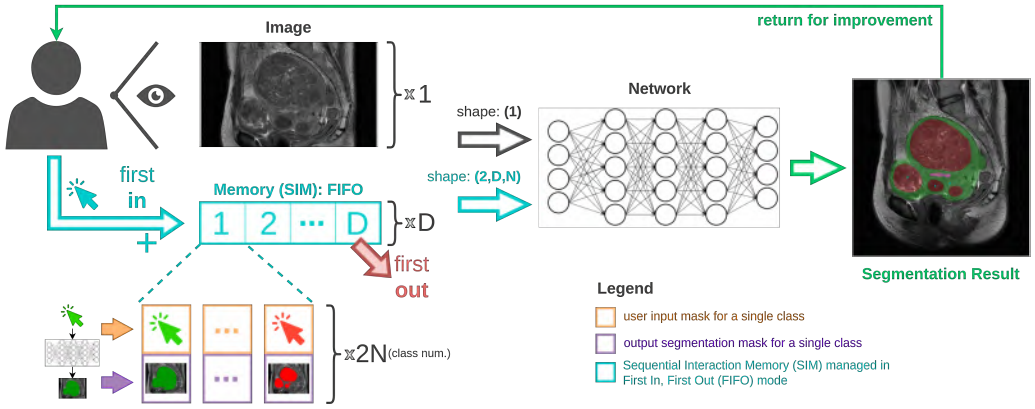

Figure 2: Proposed interactive system, featuring a network embedded in a user interaction loop and an interaction memory.

and forms an important piece of information to the system. A sequential memory was used in [53] to 'transfer' the user interaction recorded on one slice to the other slices, but was not used to exploit sequentiality during slice segmentation.

In contrast to existing work, our framework uses a sequential interaction memory which captures the sequentiality of user interactions at training and inference times. Furthermore, the proposed framework does not require specific modifications for inference and preserves low inference time. Additionally, the proposed dynamic data training specifically targets higher automatisation and generalisation at testing time by introducing a set of rules allowing for extreme variability of simulated inputs.

3. Applicative Scope

While our framework may be applied to numerous segmentation problems, we focus on the interactive slice-by-slice female pelvis MRI segmentation, involving five classes: `uterus`, `bladder`, uterine `cavity`, `tumour` and `background`. The intended use is surgical planning and surgical augmented reality [9]. We created a female pelvis MRI dataset, consisting of 97 MRI series with 3066 slices in total, manually annotated in 3D Slicer [24] and in MITK [15] by expert radiologists. This took from 10' to 50' per series with 25' on average with certain series (for instance with strong uterus deformation as in (4) in figure 1) taking more than 1 hour, which is clearly

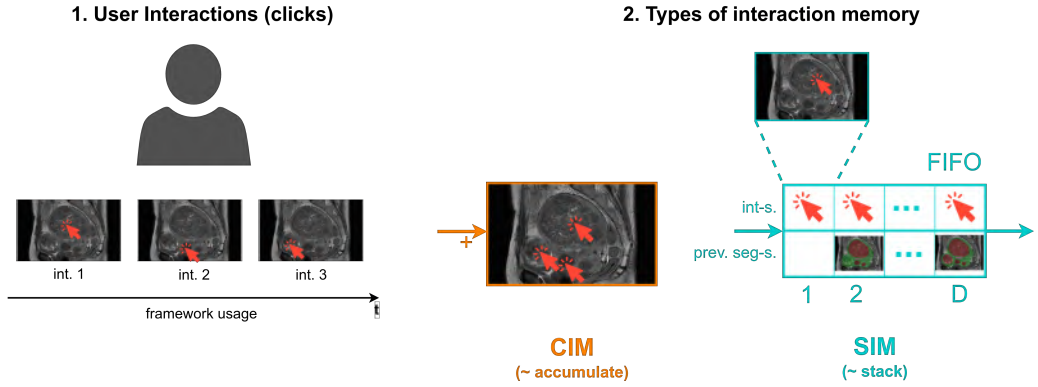

Figure 3: Interaction memory differences: (1) three individual interactions provided one-by-one with respective intermediate segmentation results obtained; (2) the Cumulative Interaction Memory (CIM) and Sequential Interaction Memory (SIM) are shown, which memorise the interactions from (1).

infeasible in the clinical setting. The segmentation of anatomical structures of the female pelvis is particularly challenging due to a large variance in their representation, including shape, size, position, orientation and texture among the patients, with and without pathologies. Moreover, it is typical for MRI data to suffer from non-uniformities of the low frequency intensity areas, which is detrimental to the network learning capabilities. Difficult samples can be seen in figure 1. On top of that, the target anatomical structures form a naturally imbalanced dataset, where `background` takes 96.15%, `uterus` 2.11%, `bladder` 1.02%, `tumour` 0.67% and uterine `cavity` 0.05%. The strongest imbalance is observed for uterine `cavity` and `background`, whose average ratio of volumes is 0.057%. The classes are also unevenly distributed throughout the dataset due to the number of the tumours varying among the series between 0 and 27. These factors further complicate learning and generally result in much lower performance on smaller classes if no mitigation against class imbalance is introduced. Our objective is to develop a segmentation system which minimises the time required to complete the segmentation with acceptable accuracy, while allowing an expert reviewer to have control and guide the segmentation, as and when necessary.

4. Methodology

We describe the system and then the training process.

4.1. System

We give the system's general structure and then the internal memory's structure.

4.1.1. Structure

We build the proposed system shown in figure 2 starting with a basic interactive segmentation system named `base`, featuring an interaction loop. This system does not have a memory of user corrections or previous segmentation results and processes each set of user corrections in isolation. The interaction loop allows iterative refinement by forming new inputs through a combination of network outputs and user corrections. The system is generic as it does not depend on a specific network architecture, as long as the network takes both the image and the user corrections as inputs. The user corrections are represented by N binary masks, where N is the number of classes. The network inputs are concatenated into a single tensor of size $H \times W \times C$, where $H \times W$ is the image size and C is the number of channels, varying depending on the system. For the base system $C_{\text{base}} = 1+N$. Indeed, as there is no memory in this system, the network takes the image as the first channel and the binary masks of the user corrections for the N classes as the next N channels. This strongly harms user experience as the past user corrections are forgotten by the system at the next interaction [50, 49].

4.1.2. Cumulative and Sequential Interaction Memory

We introduce an interaction memory, whose role is to keep track of user corrections. For that, we define a system state as a combination of user corrections and the corresponding network outputs. For the task of multi-class segmentation, a single state consists of a probability map for the network outputs and a binary mask for the user corrections, for each of the N classes. It is important to make a distinction between the interaction memory and the internal memory found in the RNN. The interaction memory tracks and stores system states, represented by inputs and outputs of the network. Indeed, the interaction memory is external to the network and does not depend on a specific network architecture. The RNN memory, however, is internal and specific to the network architecture, enabled by passing hidden states from step to step and represented by weights.

Existing works use an interaction memory, which aggregates the system states by merging the successive interaction masks [1, 52, 30]. We call this a cumulative interaction memory (CIM). The network takes the image and

the merged user correction masks, and its input tensor thus has $C_{\text{cim}} = C_{\text{base}} = 1 + N$ channels. This type of memory discards the ordering of interactions - the sequentiality, typical of user corrections. We introduce a second type of interaction memory which, in contrast to CIM, preserves the past D system states, hence the user's sequential behaviour. We call this a sequential interaction memory (SIM), and the number of states D the SIM's size or depth. The network takes an image and the SIM as inputs, which are combined to form the input tensor with $C_{\text{sim}} = 1 + 2DN$ channels. The factor 2 comes from each state containing both N interaction masks and N probability maps of intermediate segmentation results. Simply put, SIM is a container for naturally ordered input-output pairs both at training and at testing times. In other words, it is a representation of the temporal information associated with user inputs. The general differences between CIM and SIM are schematically shown in figure 3.

In our ablation study we show that RNN's suitability for sequential data may further reinforce the proposed framework. We note that the SIM does not change the system's applicability, which remains generic with respect to the data type and embedded network architecture.

4.2. Training with Dynamic Data Generation

In an interactive-loop system with an embedded network, the inputs depend on the outputs. This means that a regular training process from static data will poorly reproduce the real system usage at test time, limiting the achievable accuracy and user interaction efficiency. To resolve this, we propose a dynamic training approach, where the training data is generated from the labelled dataset during training by a virtual user. The basic idea of the virtual user is to generate a set of corrections similarly to a real user, whose involvement in training is not feasible. These corrections are represented by one binary mask per class, populated by foreground clicks for each class, including the `background` class. The click is handled by an interaction-control process, which exploits the difference image between the latest network output and the ground truth. This difference image gives a set of mislabelled regions, containing both under- and over-segmented regions. The position of the click is chosen randomly in the largest region, following a probability map whose maximum is at the region centre, decreasing towards the region boundary and vanishing outside the region. A general schematic of the Dynamic Data Generation process can be seen in figure 4. It shows an example of interaction generation for a single image containing 4 `tumour` instances

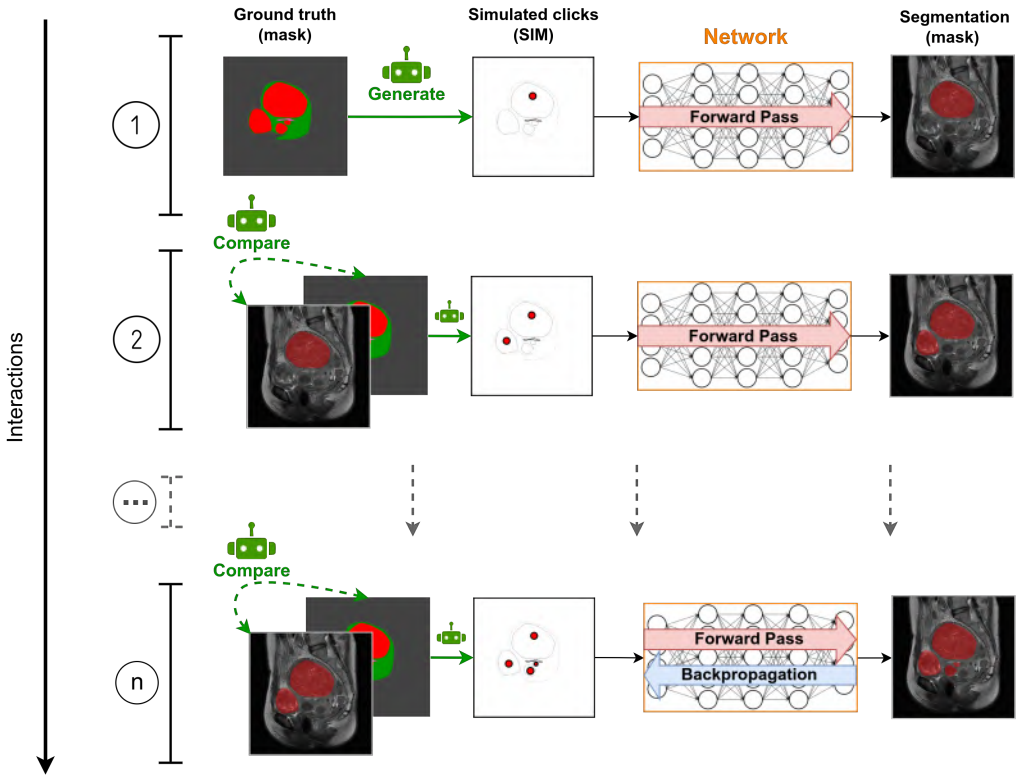

Figure 4: An example of Dynamic Data Generation (DDG) for a single input image. DDG is applied to each image each time it is encountered in the dataset. Precisely, DDG simulates a virtual user to generate the maximum of n interactions for a component of a single class. The class is represented here by 4 `tumour` instances (in red). At each interaction round, inference is performed to obtain an intermediate segmentation result, which is then compared with the ground truth to generate a new interaction based on their discrepancy. Clicks at previous interaction rounds are stored in SIM and carried over to the next round. Backpropagation is performed when all n interactions were simulated. The actions of the virtual user are marked in green. DDG is applied to all classes simultaneously following the rules in section 4.2.

with a single click generated per interaction round. In practice, the process seen in figure 4 is applied online for each image in the batch before passing on to the next batch. The standard training routine where batches are processed one-by-one is not changed, neither is any preprocessing done before the training process. There is only an additional interaction generation routine for each image.

In a typical segmentation task, each class may be represented by multiple individual components. Recall that a component is a set of spatially connected pixels pertaining to the same class in the image. When applied to our task of female pelvis MRI segmentation, this frequently occurs for all classes due to the presence of multiple instances of the same class (for the tumours) and due to the nature of the 3D MRI volume slicing (for the bladder, the uterus and the uterine cavity). For example, in certain cases the uterus' cross section may be represented in the image by multiple components due to its shape. We address this by changing how the clicks for each image are simulated and split the click simulation process in two steps. In step a), the virtual user exceptionally considers each component of each class for a potential location. In step b), the virtual user considers the mislabelled regions with larger size having higher probability of a click to be added.

In addition to interaction placement, our system implements an interaction-independence scheme, designed to ensure robustness against imperfect user behaviour at test time, with the following four main rules:

1. The maximum number of simulated interactions per component of each class is limited, typically to 3. The minimum is 0.
2. The probability of adding a subsequent interaction starts at $p \leftarrow 1$ and linearly decreases as $p \leftarrow p - \frac{1}{t}$ after each interaction round, where t is the maximum number of training interactions.
3. At each image, a random class is selected for which the user interactions are not generated.
4. A percentage of all generated interactions is held out. We typically use 80%.

These rules, along with the interaction placement control, allow the system to generate sufficiently varied interaction data throughout the training process and decrease the system's reliance on interaction supply. Specifically, rules 3 and 4 do not exist in previous work. They ensure a high level of variety in the generated data and significantly reduce interaction dependence, as evidenced by the experiments in section 5.2.3. Their rationale is threefold: 1) the framework should produce annotations for the classes not explicitly clicked on; 2) the network should consider image features instead of relying solely on user interactions; 3) the framework should be capable of automatic segmentation with no interactions provided.

Training with the proposed SIM means filling its D states with realistic values produced by the virtual user. Specifically, Dynamic Data Generation

(DDG) is the method used to form a virtual user, which generates user input artificially at training time, since the involvement of real users is not feasible. Therefore, DDG is used to fill in the sequential interaction memory during training. We thus run the system for D iterations with fixed weights to populate the SIM with simulated user input data prior to backpropagation. This is done anew each time the image is encountered in the dataset. We choose D experimentally with the goal of maximising the performance with the minimum number of interactions. At the same time, any or all of the D states may remain empty both at training and testing time to obtain a fully automatic segmentation result to be validated or subsequently refined. The DDG routine is given below as pseudo-code applicable to one specific sample image:

1. **Input** click probability p, maximum number of training interactions t
2. If $p = 1$, simulate an initial click for each component
3. If $p < 1$, simulate a corrective click for each class for the largest mislabelled region with probability p
4. Update p as $p \leftarrow p - \frac{1}{t}$
5. (rule 3) Randomly choose a class and ignore its simulated clicks
6. (rule 4) Ignore 80% of all simulated clicks
7. Form the interaction mask $\mathcal{M}$ from the simulated clicks
8. **Output** click probability p, interaction mask $\mathcal{M}$

The click probability p is managed for each image independently. It is initially set to 1 and then updated by the DDG routine.

5. Experimental Results

We describe the experiments and report the obtained results, which are then discussed in section 6.

5.1. Experimental Setup

We give implementation details and describe data augmentation and training.

5.1.1. Implementation

The proposed framework and methods are not tied to a specific network architecture. We instantiate our system with an existing encoder-decoder architecture featuring RNN modules, also called AlbuNet [42], optionally modified with LSTM layers in the decoder. Specifically, we use a ResNet34 [19] encoder and a decoder equipped with a standard convolutional layer and a matching convolutional LSTM (Long Short-Term Memory) layer at every step of the upsampling path as shown in figure 5. The reasons for which we chose this UNet are its known efficiency in the field of medical image analysis, as shown in [5, 27]. The choice of the encoder follows the same principle. However, our framework is flexible as it allows for the use of various base architectures that can accommodate an additional temporal dimension in the input image, such as a different UNet or, for example, DeepLab v3 [6]. This adaptability is a strength of our framework.

LSTMs are generally effective at processing sequences of data due to cells containing input, output and forget gates. A typical input for an LSTM network is sequential data where the order and timing of individual elements are significant. This type of data is characterized by its temporal or sequential nature, meaning that the relationship between elements depends on their position in the sequence. These properties make LSTMs beneficial for our framework, where LSTM layers reinforce sequentiality by retaining and reusing useful information about previous interactions and improve performance, as shown by the ablation study in section 5.2.1.

As compared to CIM, for which the network's input tensor has $C_{\text{cim}} = 1 + N$ channels, where N is the number of classes, with SIM, we have $C_{\text{sim}} = 1 + 2DN$ channels, where D is the SIM's size or 'depth'. The first channel is the image. The factor 2 comes from each of the D states containing both N interaction masks and N probability maps of intermediate segmentation results. For an LSTM, the input data shape could be represented as a triplet 'samples, time steps, features', which aligns well with SIM as the samples are taken as the $2N$ masks, the time steps as the D states and the features as the image. Intuitively, each time step contains a series of user interactions. The network then processes this data, learning from the sequence of features across time steps for each sample. For practical reasons, to not lose the possibility to use pre-trained encoders, we introduced LSTM layers only in the decoder, which limits the effect on performance. However, the proposed framework does not prohibit other configurations.

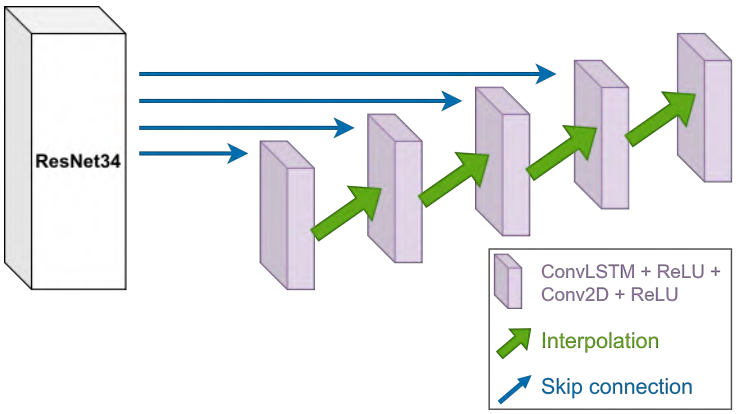

Figure 5: A general schematic of the network architecture used in the complete proposed system (`DDG-SIM`): the ResNet34 encoder pre-trained on ImageNet and a decoder with a convolutional LSTM layer at every step of the upsampling path.

The encoder was pre-trained on ImageNet [11] as a source dataset and subsequently fine-tuned on the proposed MRI female pelvis dataset without frozen layers. While the domain gap is present, transfer learning from ImageNet still proved beneficial for the stability of the training process and the final model's performance. To counter the dataset imbalance, we use the focal loss [31] and dataset-wide precalculated per-class weights.

5.1.2. Data Augmentation and Split

To avoid inter-slice and inter-patient bias, we denote a single MRI series as the smallest, undivisible element of the dataset and split the dataset as follows: the training set with 77 series containing 2449 slices, the validation set with 10 series containing 308 slices and the test set with 10 series containing 309 slices. We padded lower resolution images to 512 by 512, which is the maximum resolution of a single image for our data. We preprocessed all data via normalisation, standardisation and N4BFC [48], and performed random data augmentation: vertical and horizontal flipping, intensity shifting for brightness, gamma correction for contrast, as well as blurring and unsharp masking for sharpness adjustment.

5.1.3. Training

We trained the network on a single Nvidia P40 GPU with 24 gigabytes of video memory. The chosen batch size was 4. We employed Adam optimizer with standard parameters and a static learning rate of 0.00005. The shape

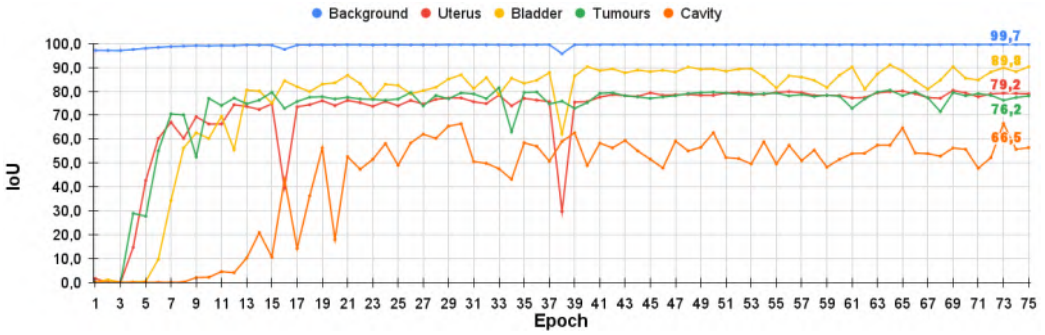

Figure 6: Performance on the validation set. The model at the 73rd epoch was chosen for the evaluation.

of a single input tensor is the shape of the SIM, which is $C_{\text{sim}} = 1 + 2DN$, where D is the memory's depth and N is the number of classes, including `background`. The network was trained for 75 epochs with the best performance on the validation set achieved at the 73rd epoch. The performance on the validation set given as IoU is shown in figure 6. It is shown that the training remains stable with the SIM as an input and the DDG training scheme.

5.2. Automated Evaluation

We give an evaluation performed automatically using the virtual user.

5.2.1. Ablation Study

We compared one automatic method and four interactive methods on the created female pelvis MRI dataset, where SDG is Static Data Generation and DDG is Dynamic Data Generation:

1. `Auto`: U-Net with ResNet34 encoder [28];
2. `SDG-base`: memory-less system trained with SDG, as described in [1];
3. `SDG-CIM`: network from `SDG-base` used with a CIM overlay;
4. `DDG-CIM`: system with CIM trained with DDG;
5. `DDG-SIM`: complete proposed system with SIM trained with DDG.

The evaluation setup uses the same network architecture, preprocessing and data augmentation across all systems with a minor network architecture change for `DDG-SIM`. `DDG-SIM` features a ResNet34 encoder with (1-4) a generic decoder or (5) an LSTM-decoder as described in section 5.1. At test time, clicks are generated via the virtual user.

Table 1: Experimental evaluation results where bold means best and underlined second best. Rows (1-9): existing methods, rows (10-14): ablation study for the proposed framework. `GrabCut` [38], `VMN`, `NuClick` and `BRS` versions are used per-class, hence `background` metrics are not provided.

Method ↓	Background		Uterus		Bladder		Tumours		Cavity	
	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
GrabCut	-	-	17.6	25.1	14.8	21.0	21.7	29.8	8.0	12.4
VMN	-	-	57.6	72.0	78.3	86.1	42.6	55.7	19.8	27.4
NuClick	-	-	23.6	33.1	41.3	54.9	47.0	55.5	<u>52.2</u>	<u>67.7</u>
NoBRS	-	-	36.7	49.7	20.7	30.7	32.4	44.1	7.4	12.1
BRS	-	-	37.4	50.5	21.5	31.6	33.1	44.8	7.8	12.6
RGB-BRS	-	-	37.5	50.6	21.6	31.7	33.1	44.9	7.8	12.6
f-BRS-A	-	-	37.3	50.5	23.9	32.0	33.3	45.1	7.7	12.4
f-BRS-B	-	-	38.3	51.6	23.1	33.3	33.8	45.4	9.6	14.6
f-BRS-C	-	-	37.5	50.7	21.7	31.8	33.0	44.8	7.9	12.7
Auto	99.2	99.6	64.7	78.6	71.9	83.6	60.4	75.3	40.4	57.6
SDG-base	99.1	99.6	61.7	76.3	70.1	82.4	62.5	76.9	21.1	34.9
SDG-CIM	99.3	99.7	66.5	79.9	83.9	91.2	72.8	84.3	29.0	44.9
DDG-CIM	<u>99.6</u>	<u>99.8</u>	<u>77.4</u>	<u>87.3</u>	**87.4**	**93.3**	<u>77.7</u>	<u>87.4</u>	39.6	56.7
DDG-SIM	**99.6**	**99.8**	**79.8**	**88.7**	<u>87.0</u>	<u>93.0</u>	**79.0**	**88.3**	**57.8**	**73.3**

5.2.2. Comparison with State-of-the-Art

We compared our framework with two classical interactive methods and eight interactive deep learning methods on the created female pelvis MRI dataset:

1. `VMN`: volumetric memory network trained with SDG, as described in [53] and inputting extreme clicks;
2. `NuClick`: a segmentation network introduced for microscopy images and trained dynamically in [26];
3. `BRS`: a backpropagating refinement scheme for mislabeled locations correction, training-less by definition, in [23];
4. `RGB-BRS`: `BRS` minimised with respect to the RGB image instead of distance maps in [46];
5. `f-BRS variants`: improved `BRS`, `f-BRS` solves an optimization problem with respect to auxiliary variables instead of the network inputs as in `BRS`
   (a) `f-BRS-A`: introduces scale and bias after the backbone

(b) `f-BRS-B`: introduces scale and bias before the first separable convolutions block in DeepLabV3+ [7]
(c) `f-BRS-C`: introduces scale and bias before the second separable convolutions block in DeepLabV3+ [7]
(d) `NoBRS`: using network architecture from [23] without `BRS`.

For `VMN` [53], `BRS` variants [23, 46] and `NuClick` [26] we use the code made publicly available by the authors and recommended parameters. We trained `VMN` [53] on our dataset, reducing the batch size to 4 to keep the computation overhead feasible. For the `BRS variants` [23, 46] and `NuClick` [26] we used pre-trained models made publicly available by the authors. The models are *resnet34_dh128_sbd* and *NuClick_Nuclick_40xAll* respectively. The metrics are reported in table 1.

5.2.3. Click Number Influence

An interactive segmentation system refines the segmentation result via user interactions. In essence, this is inputting clicks into the system to provide additional information. Hence, the number of clicks is a key influencing factor in the framework's performance.

We perform a systematic evaluation of the influence of the number of clicks at train and test time on the segmentation accuracy. For this, human user involvement is not feasible due to the number of series and the need to re-segment them for each evaluation setting. Therefore, we perform this evaluation as in section 5.2.1 via the virtual user generating simulated interactions at test time. `DDG-SIM` is used for the evaluation, where we control only two parameters: the number of clicks simulated at train and at test time. Three setups are provided: (1) training - fixed maximum click number, testing - varying click number; (2) training, testing - equal click number; (3) training - `Auto`, default `DDG-SIM`, modified `DDG-SIM` with rules 2-4 from section 4.2 disabled, testing - 0 clicks. The purpose of these setups is as follows: (1) evaluate the influence of the number of clicks at testing on the performance; (2) evaluate the influence of the number of clicks at training on the performance; (3) evaluate the performance of `DDG-SIM` when no clicks are provided with and without rules 2-4 from section 4.2, which should improve the system's ability to automatically segment regions.

For setup (1), the maximum number of clicks simulated at train time is fixed to 3 which is the default value for `DDG-SIM`, while the number of clicks at test time varies. We then report IoU for all classes when simulating 0 (`Auto`), 1, 2, 3 and 6 clicks at testing in figure 7.

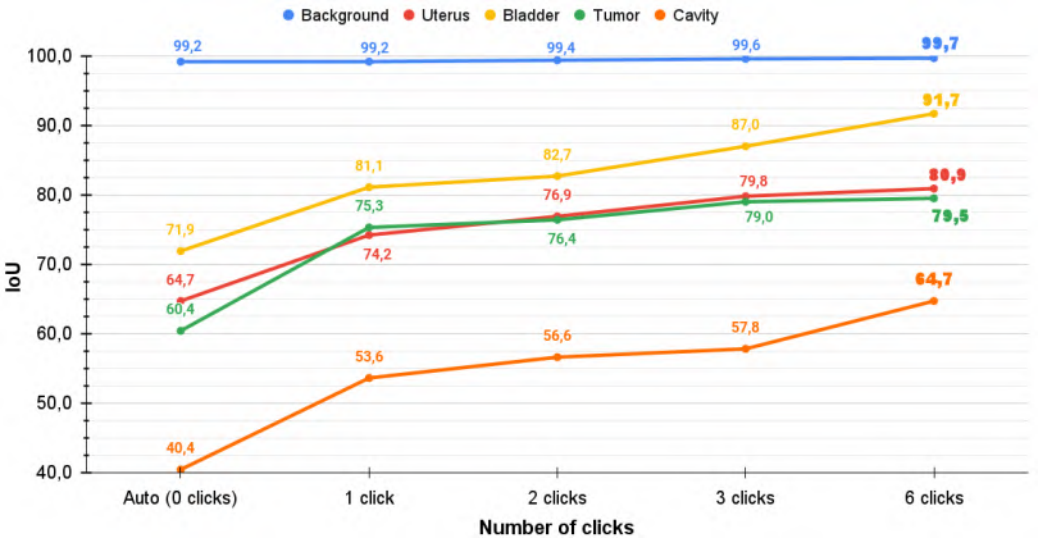

Figure 7: `DDG-SIM`: Influence of the number of clicks simulated at test time on the IoU score, compared with `Auto` (0 clicks). The strongest improvement presents itself at the first click. Bold means best.

For setup (2) we fix the number of clicks simulated both at train and test time so that they are equal (such as a maximum of 3 clicks at training and exactly 3 clicks at testing) and change them jointly. We then report the IoU for all classes when simulating 0 (`Auto`), 1, 2, 3 and 6 clicks in figure 8.

For setup (3), we evaluate `DDG-SIM` performance when providing no clicks at testing. We compare `Auto`, default `DDG-SIM` and modified `DDG-SIM` with rules 2-4 from section 4.2 disabled. We report IoU for all classes in figure 9.

5.2.4. Generalisation Study

We further evaluate the complete proposed system `DDG-SIM` on two other tasks with different modality and objects of interest - namely, on liver and pancreas CT segmentation. We use the "Liver Tumours" and "Pancreas Tumour" medical segmentation decathlon datasets [45] and compare our framework's performance on these data to the methods participating in the corresponding challenge [2] as well as `VMN` [53]. Each of the datasets was initially assembled for the task of multi-class segmentation with liver CT targets being `liver` and `cancer`, and pancreas CT targets being `pancreas` and `mass` (cyst or tumour). While this challenge is aimed at automatic segmentation approaches, a comparison with interactive methods may further prove their

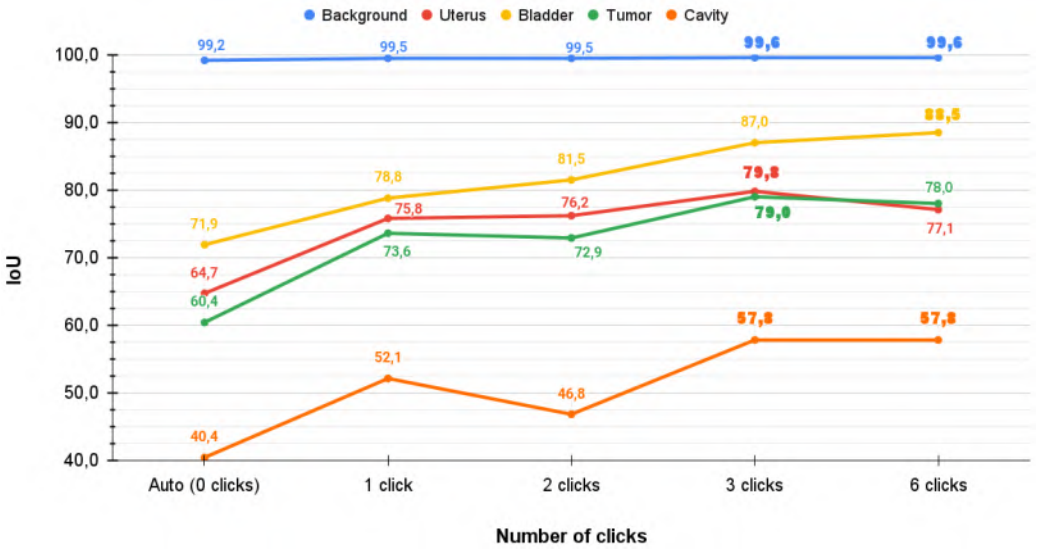

Figure 8: `DDG-SIM`: Influence of the number of clicks simulated at train time on the IoU score, compared with `Auto` (0 clicks). The number of clicks simulated at training and at testing are equal and change jointly. The overall performance improvement is less noticeable after 3 clicks. Simulating 3 clicks at training is our choice for `DDG-SIM` with the current data. Bold means best.

feasibility for the tasks usually requiring expert's validation and potential refinement. For `VMN` [53] we use the code made publicly available by the authors and recommended parameters on these new datasets. We reduce the batch size to 4 due to the limited GPU availability.

The ground truth labels for the test set were not made available for this challenge. We thus randomly split the publicly available training sets for both `liver` and `pancreas`, using approximately 70%/15%/15% for training, validation and test respectively. As a result, the split is 91/20/20 series for the liver and 198/42/42 series for the pancreas datasets. Effectively, this means that the training is performed on much lower-size datasets than those of the competing methods, which makes it more challenging. To add to this, the key difficulty of these datasets is label imbalance with both large (`liver`, `pancreas`) and small (`mass` or `cancer`) targets. The metrics are reported in figure 10 for both `liver` and `pancreas`.

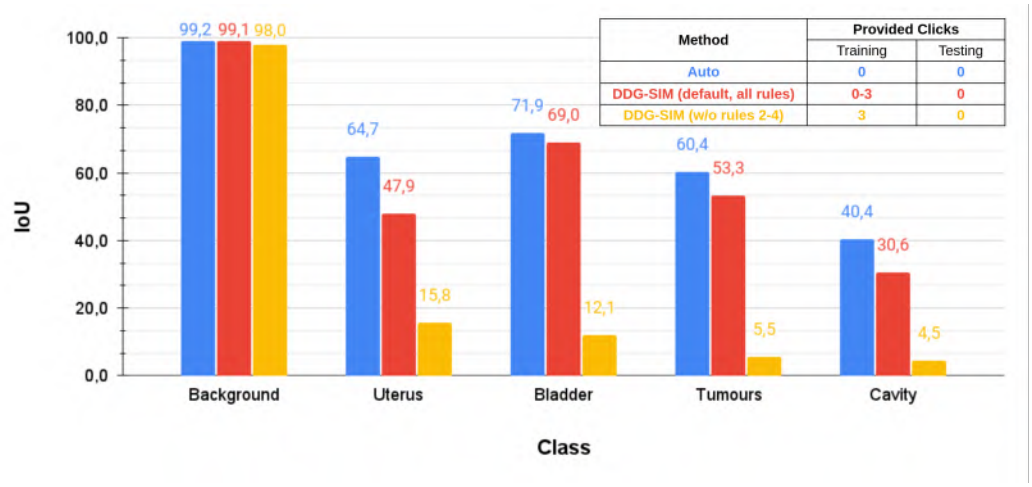

Figure 9: `DDG-SIM`: Performance with 0 clicks provided at testing. `Auto`, default `DDG-SIM` and modified `DDG-SIM` (with rules 2-4 from section 4.2 disabled) are compared. Disabling the rules makes automatic segmentation fail. This illustrates the automatic segmentation capability of `DDG-SIM` as brought by the DDG training process and hence the importance of having varied interaction data when simulating clicks.

5.3. User Evaluation

We performed a user study with `DDG-SIM` involving eight medical experts, using a specifically developed graphical user interface. All experts have a background in gynaecology, with one exception, a radiologist with specialisation in urology and junior experience level. For clarity, we assign a letter and a number to each expert as follows: A - senior gynaecology surgeon; B1-2 - senior radiologists; C - junior gynaecology surgeon; D1-3 - junior radiologists; E - junior urologic surgeon with experience in gynaecology.

We randomly selected 6 test series containing 144 slices in total, where 1 series is used to familiarise the users with the graphical user interface and 5 series are used in a random order for user evaluation. MRI image samples from each of the series can be seen in figure 1. We evaluate the user performance in figure 13 using mIoU per series for each expert in comparison to the `Auto` method as in section 5.2. In the same manner, the elapsed time is compared in figure 12. The segmentation results are compared with the `Auto` method in figure 11. Figure 14 shows mIoU over each class per series.

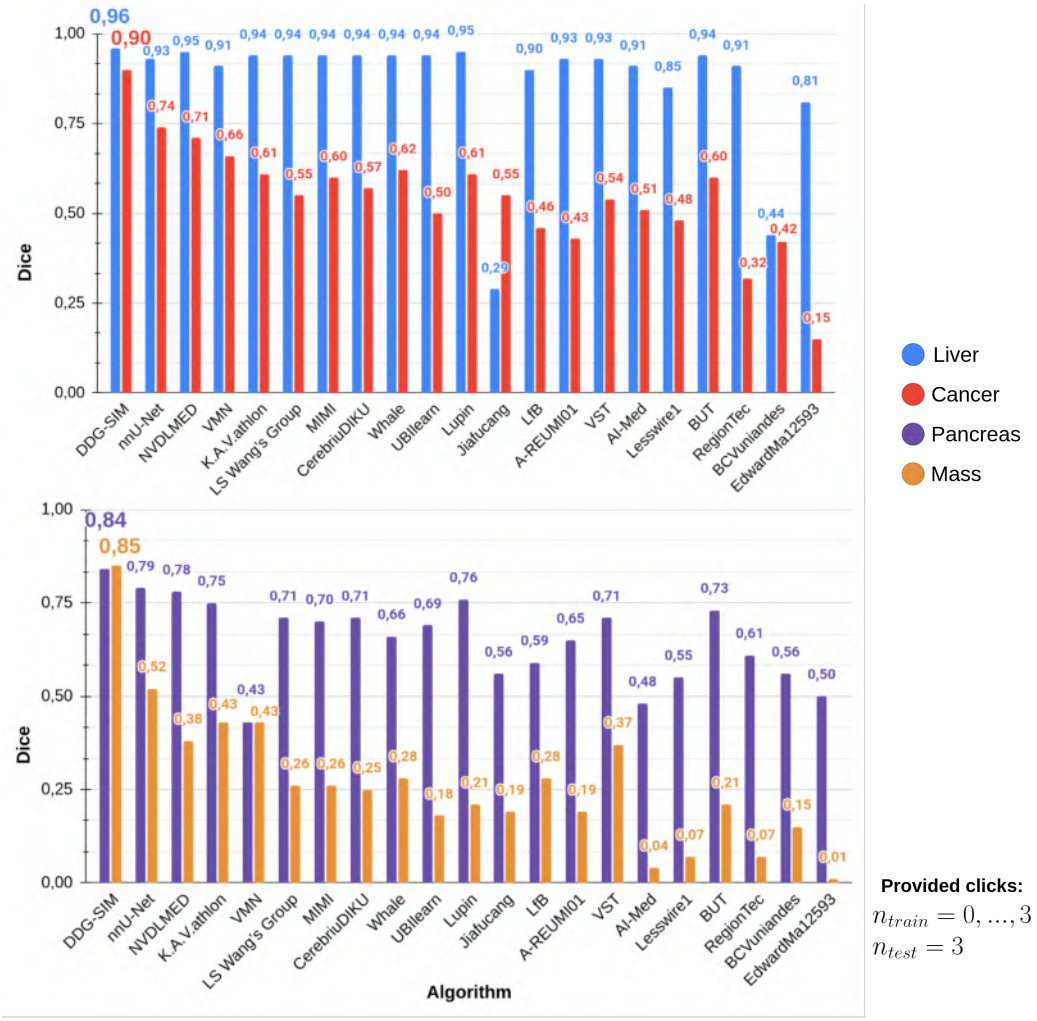

Figure 10: `DDG-SIM` experimental evaluation results given as Dice on the medical segmentation decathlon "Liver Tumours" (`liver` - blue, `cancer` - red) and "Pancreas Tumour" (`pancreas` - purple, `mass` - orange) datasets in comparison to the automatic segmentation approaches participating in the challenge, where bold means best. `VMN` is a state of the art interactive segmentation approach. The number of simulated clicks is provided for both training and testing in the bottom-right hand corner.

5.4. Inference Time Analysis

We report the average inference time for a single image and compare it with those of the existing interactive segmentation approaches in table 2.

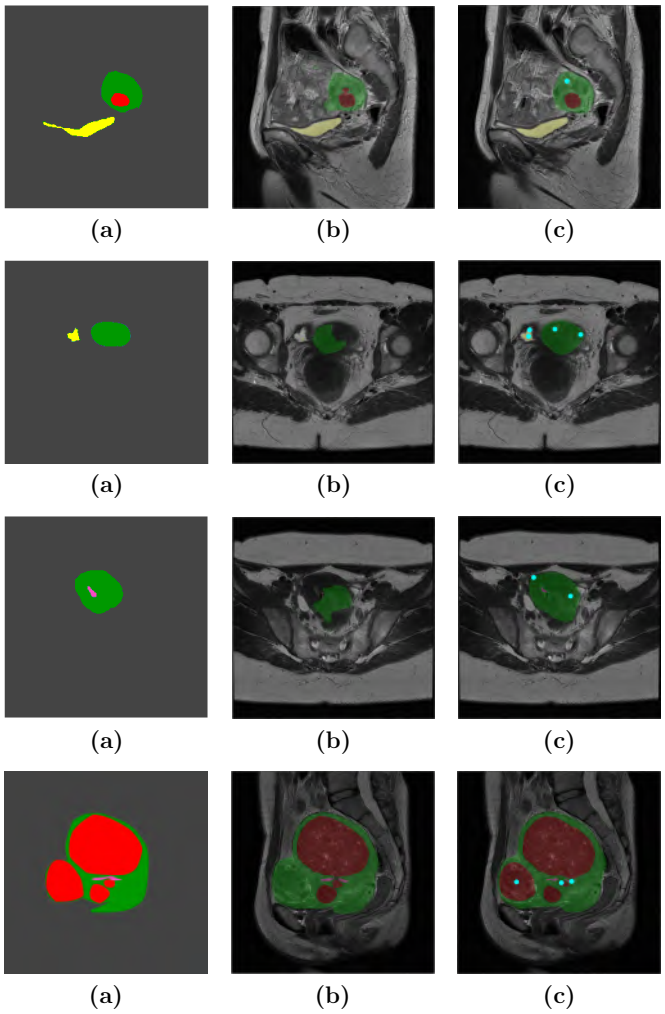

Figure 11: Segmentation results, where `uterus` - green, `bladder` - yellow, `tumour` - red, `cavity` - pink and user clicks - cyan: **(a)** ground truth; **(b)** `Auto`; **(c)** human user-controlled `DDG-SIM`.

6. Discussion

We discuss the results obtained in the previous section.

6.1. Automated Evaluation

We discuss results obtained with the virtual user.

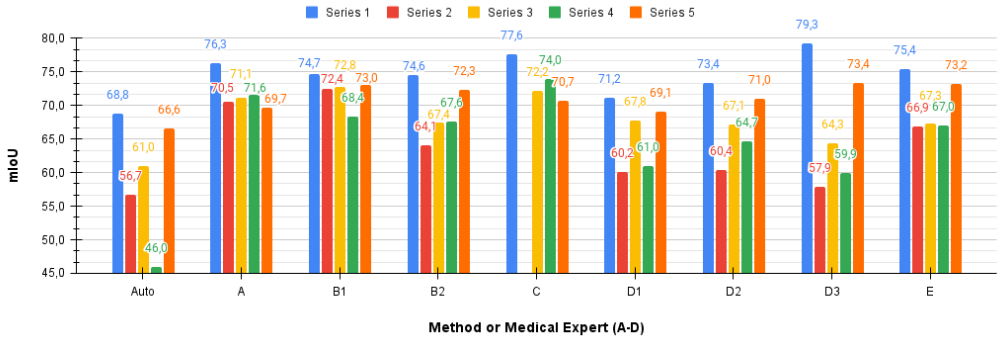

Figure 12: User Evaluation: mIoU over all classes per medical expert per series.

6.1.1. Ablation Study

The metrics are reported in table 1, where we observe that the IoU and Dice are in agreement. They show that `DDG-SIM` outperforms, with a substantial margin for `cavity`, a significant margin for `uterus` and `tumour`, a similar result for `background`, and a slight disadvantage for `bladder`, for which `DDG-CIM` slightly outperforms at 87.4% against 87.0% IoU. This demonstrates the robustness of the proposed framework. The ablation study shows a steady increase in performance, starting with `SDG-base` and adding the proposed components towards `DDG-SIM`. `Auto` outperforms both `SDG-base` on `uterus`, `bladder` and `cavity`, and `SDG-CIM` on `cavity`. This can be attributed to static data generation, which does not perform well for smaller numbers of interactions. In our experience, the higher the number of interactions at training, the lower the effectiveness of individual interactions at test time. While the opposite is also true, it can be observed from the results that certain systems may not be able to learn efficiently from a small number of interactions at training. We observe a comparatively lower accuracy for `cavity`, whose IoU lies between 21.1% and 57.8%. We explain this with its low volume, which accounts for only 0.054% of the dataset.

Examining other existing methods, this is also true for `VMN`, which achieves a good performance on `bladder`, but struggles with the more difficult classes. We find that this might be additionally due to the low number of slices in a standard female pelvis MRI scan, where the classes such as `tumour` or `cavity` may be found only on a single slice out of the whole volume in addition to occupying just a few pixels, which may interfere with the approach. Still, `VMN` shows a notable performance on `bladder`, with an IoU of 78.3%, which

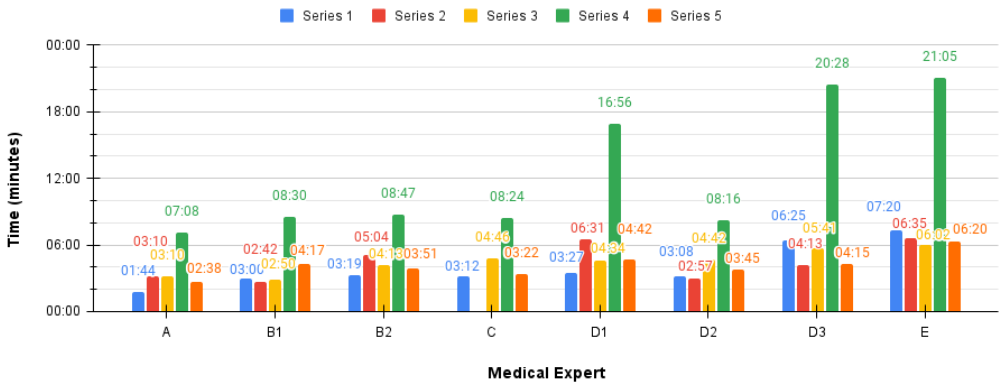

Figure 13: User Evaluation: Segmentation time per medical expert per series.

is competitive but still falls short of DDG-CIM's 87.4%. However, it struggles significantly across the other categories, particularly with cavity, where it is greatly outperformed by DDG-SIM's superior IoU of 57.8%.

Interestingly, NuClick demonstrates a notably high performance in segmenting the cavity class with an IoU of 52.2%. However, it still falls short when compared to DDG-SIM, which achieves an IoU of 57.3% in the same class. The relatively high performance of NuClick in cavity segmentation may be associated with its design and optimization for microscopy image segmentation tasks. The visual characteristics of cavity regions in such images may be similar to those that NuClick was specifically intended to segment, possibly contributing to its success in this particular class.

The BRS and f-BRS variants display a range of results, with none matching the DDG-SIM scores. The f-BRS-A, f-BRS-B, and f-BRS-C methods also fall short, with the highest IoU among them for cavity being only 9.6%, indicating a substantial gap when compared to DDG-SIM. Overall, the superiority of DDG-SIM proves it to be a solid segmentation framework in view of the state of the art.

6.1.2. Click Number Influence

Three setups are provided: (1) training - fixed maximum click number, testing - varying click number; (2) training, testing - equal click number; (3) training - Auto, default DDG-SIM, modified DDG-SIM with rules 2-4 from section 4.2 disabled, testing - 0 clicks.

Setup (1). We report IoU for all classes when simulating 0 (Auto), 1,

2, 3 and 6 clicks at testing in figure 7. The metrics show a substantial improvement of the segmentation accuracy against `Auto` when at least 1 click is provided. Furthermore, a notable growth of accuracy is also observed for all classes when transitioning to 2 clicks. At the same time, while there is a further regular improvement for `cavity` and `bladder` beyond 2 clicks, the other classes improve only slightly. This can be explained by two factors. First, a single provided click produces an IoU score close to the upper performance boundary achieved by the proposed framework, as seen in the ablation study. This does not leave much room for improvement with a given training set size (77 series, 2449 slices). Second, during training, clicks are currently simulated with a maximum of 3 for all systems. This is done to minimise the amount of interaction required from a human user at test time. The proposed DDG scheme brings the average number of simulated clicks at training even lower, which contributes to the performance stabilising below the maximum click threshold. While still limited by the current performance ceiling, increasing the maximum number of simulated clicks to 6 per class during training may allow to achieve a more stable performance growth with each added click at testing time. At the same time, with 6 clicks a human user evaluation experience would be negatively affected. Indeed, each individual click would bring less improvement, generally requiring more clicks for the same task, which is undesirable in a clinical setting.

Setup (2). We report the IoU for all classes when simulating 0 (`Auto`), 1, 2, 3 and 6 clicks in figure 8. The metrics show a substantial improvement of the segmentation accuracy against `Auto`, demonstrating robustness of `DDG-SIM` for any number of clicks at training. The strongest performance improvements are observed between `Auto` and training `DDG-SIM` with 1 click, as well as between training `DDG-SIM` with 1 click and with 3 clicks. Performance with 2 clicks shows an overall improvement over that with 1 click, but `cavity` and `tumour` classes show notable and slight performance decrease respectively. This can be explained by the dynamic data generation rules we use described in section 4.2, which target the increase of individual click efficiency. Specifically, rules 2-4 are such that with the chosen maximum of 1 click at training, it is often the case that no clicks will be simulated at all for many of the labels. This makes the system more reliant on the underlying image features, which places it closer to `Auto`, but still provides a significant performance improvement due the interactivity. In contrast, the chosen maximum of 3 clicks at training allows for more consistent click simulation, which significantly improves performance. At the same time, the maximum

of 2 clicks at training is an in-between case, where the actual simulated click number does not seem to be sufficient for the `cavity` and `tumour` labels, which are represented by multiple blobs of varying size and clarity. In this case, clicks are simulated only for a small number of these blobs (such as for 1 out of the 7 tumours in a single slice), which does not allow for consistent learning from clicks and reduces the performance on these classes.

Setup (3). We report IoU for all classes in figure 9. The metrics show that when providing no interactions, default `DDG-SIM` significantly outperforms modified `DDG-SIM`, where the chosen maximum number of clicks was consistently simulated for each class during training. Specifically, default `DDG-SIM` and modified `DDG-SIM` are respectively 50% against 9% in terms of the mIoU score (`background` class excluded). Simply put, this figure shows that the use of DDG allows our framework, when used without user interactions, to obtain performance comparable with state-of-the-art fully-automatic segmentation. It also shows that, should the proposed rules 2-4 of DDG were disabled, the framework would fail to perform any meaningful segmentation without user interactions, indicating strong dependence on the number and exhaustivity of interactions provided at training time. Clearly, the more interactions are provided at training time, the lesser is the network's capability for automatic segmentation in general and for segmentation of unclicked components in particular. Intuitively, if interactions are scarce, the network focuses more on image features, resulting in higher automation at testing time. While `Auto` with 59% mIoU outperforms both default and modified `DDG-SIM` when no clicks are provided, the interactive approaches accuracy can be improved further with additional clicks as shown in figure 8, which is not the case for `Auto`.

6.1.3. Generalisation Study

The metrics are reported in figure 10 for both `liver` and `pancreas`. They show that the proposed interactive framework outperforms the best automatic methods on all classes, with a substantial margin for liver `cancer` and pancreas `mass` - 90% against 74% and 85% against 52% respectively and a slight advantage for `liver` and `pancreas` classes - 96% against 95% and 84% against 79% respectively. This shows that the proposed framework is generically applicable to segmentation tasks and data types.

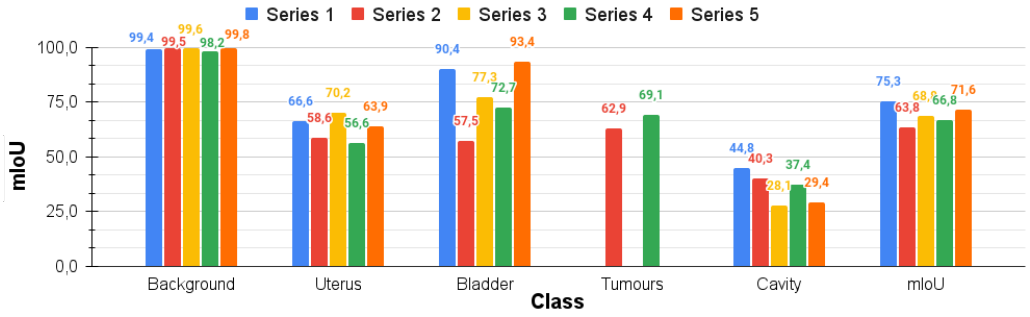

Figure 14: User Evaluation: mIoU over all medical experts per class per series.

6.2. User Evaluation

The elapsed annotation time per series is compared in figure 12. We note that the segmentation time is low enough to be clinically feasible, even if the users are barely acquainted with the system. Indeed, the average elapsed time for all series is 5'56", which is largely below the reported average of 25' for existing systems. Series 4 was a complex case with 11 tumours and a heavy deformation of the uterus shape, taking 12'26" on average for our system and more than 40' for existing systems. Furthermore, as seen in figure 1, each of the series used for user evaluation is challenging in its own manner. While the proposed framework facilitates the segmentation process, interpretation of the MRI images by the human user remains a task in itself. This explains the elapsed time and the mIoU score discrepancies between senior and junior experts, especially noticeable for series 4 with the peak of 20'28" for junior experts, 8'47" for senior experts in gynaecology and 21'05" for E, the junior expert in urology. Figure 13 shows that our framework substantially outperforms automatic segmentation on all data with a lesser improvement for series 5. This is especially noticeable for the difficult series 4, which achieved a score of 46,0% for `Auto` against the average of 66,7% over all interactive human-guided segmentations. While expert E mainly specialises in urology, only the segmentation time appears to be affected, thus attributed to an increased difficulty in image interpretation. Still, the segmentation accuracy of expert E is on par with the other experts.

Figure 14 shows mIoU over each class per series. We observe a comparatively low accuracy of `cavity` segmentation during user evaluation, similarly to the automated tests. This is because of the small size of the cavity and its lack of clear outer contours. In addition, the slices may split the cavity in a

Table 2: Reported average inference time and standard deviation for DDG-SIM in comparison to existing interactive segmentation approaches, where bold means best and underlined second best.

Method	Inference time (ms)
BRS [22]	810
Interactive 3D nnU-Net [20]	500
IteR-MRL [30]	470
f-BRS-B [46]	226
FocusCut [32]	118
FocalClick B0-S1 (on CPU) [8]	100
VMN [53]	53
DDG-SIM (ours)	$\underline{47.2 \pm 6.2}$
[40]	**40**

manner that makes it appear in several isolated small components, in which case some components may be ignored by the users. This creates a variability in the dataset and a potentially large discrepancy between the ground truth and the user segmentation. While this is typical for other female pelvis MRI objects of interest, the cavity's small size strongly amplifies any slight segmentation discrepancy.

6.3. Inference Time Analysis

As seen in table 2, our framework with $47.2 \pm 6.2ms$ per image is on par or significantly faster than existing methods. This amounts to approximately 21 FPS, which is well adapted for an interactive clinical application. Hence, usage of SIM with here up to 5 classes does not introduce significant overhead and leaves sufficient room for additional computational complexity (e.g. additional classes or a deeper network).

6.4. Implications and Limitations

On the most general level, we find that temporal information associated with user interactions is overlooked in existing methods. Simply put, CIM, which is used in most previous works, does not convey the sequential nature of interactions, discarding the temporal component naturally present in the way the user interacts with the annotation software. However, the proposed SIM conveys this information and its use improves segmentation performance. Furthermore, DDG during training has a significant impact not only on the

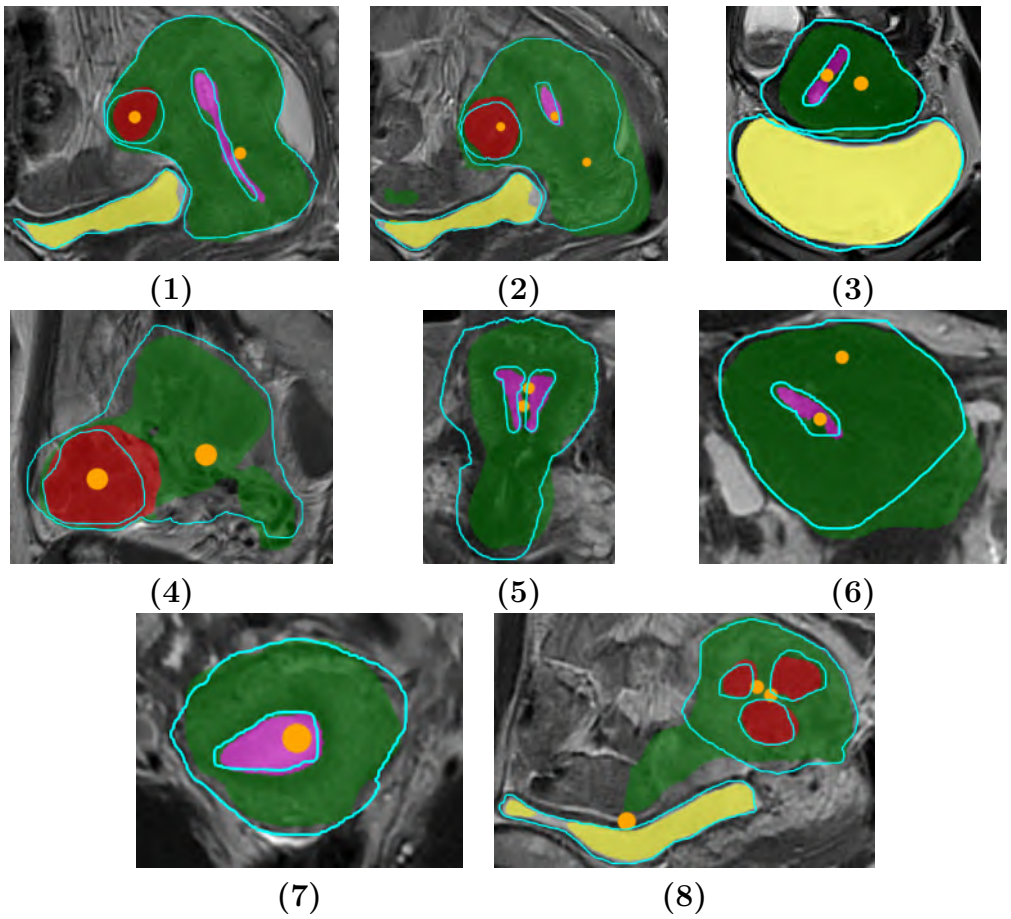

Figure 15: Segmentation failure cases, where `uterus` - green, `bladder` - yellow, `tumour` - red, `cavity` - pink, user clicks - orange and ground truth - cyan: **(1,7)** most widespread case with contours having a slight divergence with the ground truth; **(2-5,7-8)** under-segmentation; **(2,4-5,6-8)** over-segmentation. The maximum number of clicks is fixed to three. Additional clicks allow to notably reduce under- and over-segmentation, resulting in segmentations comparable to **(1,7)**. The metrics for `cavity`, present in **(1-3,5-7)**, are affected most strongly in all cases due to its size.

method's performance, but also on the user experience. Specifically, the ensemble of interaction generation rules in section 4.2 allows the network to produce automatic segmentations comparable to fully-automatic methods without user interactions, as well as to segment most of the objects of interest at once by providing a single click for any one of them. This has a large impact for the time-constrained clinical environment.

We show segmentation failure cases in figure 15. From our experiments we observe that additional clicks allow to reduce under- and over-segmentation until a case similar to cases 1 and 7 in figure 15 is reached. However, remaining divergence from the ground truth notably affects `cavity` metrics due to `cavity`'s size.

One limitation of our method is the impact on training speed. The necessity to populate the sequential memory by doing multiple inferences increases the time to process each image. However, the inference time being 47.2 ± 6.2 ms, the training time remains reasonable, even with multiple additional inferences per image. Another constraint is the sequential memory size - increasing memory size C_{sim} increases the computational complexity, especially when using LSTM blocks. However, making the memory too large seems to be counter-intuitive, since the interest lies in having the minimal number of clicks required for a high-quality segmentation at testing, which implies limiting the number of clicks at training and hence C_{sim} in some manner. We show experimentally that, in most cases, providing more than 3 clicks has diminishing returns, and 3 or less clicks produce the most important improvement, suggesting that large C_{sim} is actually counter-productive.

The proposed framework utilises a parameter for the maximum number of clicks, which serves as a starting point for the dynamic data generation described in section 4.2. In our experiments we extensively show that 3 or less clicks produce the results surpassing those of the comparable frameworks on multiple tasks. However, one can imagine that the maximum number of clicks may change depending on the task, making it a parameter to tune, which might be undesirable if more automation is desired. For this, it might be of interest to select it in an automatic manner for each image in future work. For example, simply by calculating the metrics for each click generation at training time and decreasing the probability of adding a new click along with the increase in accuracy.

7. Conclusion

We have proposed a general deep learning-based interactive multi-class image segmentation framework, with a user interaction loop and a sequential interaction memory. The embedded network is trained on dynamically generated data to improve performance and reduce interaction-dependence. We have demonstrated our framework in female pelvis MRI segmentation, using a new dataset. Furthermore, we successfully applied it to the tasks of

`liver` and `pancreas` CT segmentation from the medical segmentation decathlon challenge, showing the best overall performance. We have evaluated our framework against existing work in an ablation study with the standard metrics, observed the influence of the number of interactions at test time on performance and conducted a user evaluation, involving 8 medical experts with gynaecology background and varying experience levels to use our software via a specifically-developed GUI. This shows that our framework largely outperforms existing systems in accuracy and drastically reduces the average user segmentation time from 25' to 5'56" when used by either senior or junior experts.

We plan to further improve the proposed solution towards its clinical usage. First, through application to other segmentation tasks and expansion of the user study. Second, by using it to aid annotation, reducing interaction demand through SIM initialisation with automatic segmentation. Third, by expanding the neural network's direct inputs to 3D images, to further shorten the segmentation time thanks to the natural dependencies that exist between the multiple slices.

References

[1] Amrehn, M., Gaube, S., Unberath, M., Schebesch, F., Horz, T., Strumia, M., Steidl, S., Kowarschik, M., Maier, A., 2017. UI-Net: Interactive Artificial Neural Networks for Iterative Image Segmentation Based on a User Model, in: Eurographics Workshop on Visual Computing for Biology and Medicine, The Eurographics Association.

[2] Antonelli, M., Reinke, A., Bakas, S., Farahani, K., AnnetteKopp-Schneider, Landman, B.A., Litjens, G.J.S., Menze, B.H., Ronneberger, O., M.Summers, R., van Ginneken, B., Bilello, M., Bilic, P., Christ, P.F., Do, R.K.G., Gollub, M.J., Heckers, S., Huisman, H.J., Jarnagin, W.R., McHugo, M., Napel, S., Pernicka, J.S.G., Rhode, K.S., Tobon-Gomez, C., Vorontsov, E., Meakin, J.A., Ourselin, S., Wiesenfarth, M., Arbeláez, P., Bae, B., Chen, S., Daza, L.A., Feng, J.J., He, B., Isensee, F., Ji, Y., Jia, F., Kim, N., Kim, I., Merhof, D., Pai, A., Park, B., Perslev, M., Rezaiifar, R., Rippel, O., Sarasua, I., Shen, W., Son, J., Wachinger, C., Wang, L., Wang, Y., Xia, Y., Xu, D., Xu, Z., Zheng, Y., Simpson, A.L., Maier-Hein, L., Cardoso, M.J., 2022. The medical segmentation decathlon. Nature Communications 13.

[3] Boykov, Y., Jolly, M.P., 2001. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images, in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, pp. 105–112 vol.1.

[4] Cardenas, C.E., Yang, J., Anderson, B.M., Court, L.E., Brock, K.B., 2019. Advances in Auto-Segmentation. Seminars in Radiation Oncology 29, 185–197. doi:https://doi.org/10.1016/j.semradonc.2019.02.001.

[5] Chaisangmongkon, W., Chamveha, I., Promwiset, T., Saiviroonporn, P., Tongdee, T., 2021. External validation of deep learning algorithms for cardiothoracic ratio measurement. IEEE Access 9, 110287–110298. doi:10.1109/ACCESS.2021.3101253.

[6] Chen, L.C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. ArXiv abs/1706.05587.

[7] Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation, in: European Conference on Computer Vision.

[8] Chen, X., Zhao, Z., Zhang, Y., Duan, M., Qi, D., Zhao, H., 2022. Focalclick: Towards practical interactive image segmentation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 1290–1299.

[9] Collins, T., Pizarro, D., Gasparini, S., Bourdel, N., Chauvet, P., Canis, M., Calvet, L., Bartoli, A., 2021. Augmented reality guided laparoscopic surgery of the uterus. IEEE Transactions on Medical Imaging 40, 371–380.

[10] Criminisi, A., Sharp, T., Blake, A., 2008. Geos: Geodesic image segmentation, in: Forsyth, D., Torr, P., Zisserman, A. (Eds.), Computer Vision – ECCV 2008, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 99–112.

[11] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.

[12] Diaz-Pinto, A., Alle, S., Ihsani, A., Asad, M.H., Nath, V., P'erez-Garc'ia, F., Mehta, P., Li, W., Roth, H.R., Vercauteren, T.K.M., Xu, D., Dogra, P., Ourselin, S., Feng, A., Cardoso, M.J., 2022. Monai label: A framework for ai-assisted interactive labeling of 3d medical images. ArXiv abs/2203.12362.

[13] Futrega, M., Milesi, A., Marcinkiewicz, M., Ribalta, P., 2021. Optimized u-net for brain tumor segmentation. ArXiv abs/2110.03352.

[14] Ghiasi, G., Lin, T.Y., Pang, R., Le, Q.V., 2019. Nas-fpn: Learning scalable feature pyramid architecture for object detection. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 7029–7038.

[15] Goch, C.J., Metzger, J., Nolden, M., 2017. Abstract: Medical research data management using mitk and xnat, in: Maier-Hein, geb. Fritzsche, K.H., Deserno, geb. Lehmann, T.M., Handels, H., Tolxdorff, T. (Eds.), Bildverarbeitung für die Medizin 2017, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 305–305.

[16] Grady, L., 2006. Random walks for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1768–1783.

[17] Guo, Y., Gao, Y., Shen, D., 2016. Deformable mr prostate segmentation via deep feature learning and sparse patch matching. IEEE Transactions on Medical Imaging 35, 1077–1089.

[18] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H., 2017. Brain tumor segmentation with deep neural networks. Medical Image Analysis 35, 18–31.

[19] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 770–778.

[20] Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P.F., Kohl, S.A.A., Wasserthal, J., Koehler, G., Norajitra, T., Wirkert, S.J., Maier-Hein, K., 2018. nnu-net: Self-adapting framework for u-net-based medical image segmentation. ArXiv abs/1809.10486.

[21] Jahanifar, M., Tajeddin, N.Z., Koohbanani, N.A., Rajpoot, N.M., 2021. Robust interactive semantic segmentation of pathology images with minimal user input. 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) , 674–683.

[22] Jang, W.D., Kim, C.S., 2019a. Interactive image segmentation via backpropagating refinement scheme, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5292–5301. doi:10.1109/CVPR.2019.00544.

[23] Jang, W.D., Kim, C.S., 2019b. Interactive image segmentation via backpropagating refinement scheme. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 5292–5301.

[24] Kikinis, R., Pieper, S.D., Vosburgh, K.G., 2014. 3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support. Springer New York, New York, NY. pp. 277–289.

[25] Kline, T.L., Korfiatis, P., Edwards, M.E., Blais, J.D., Czerwiec, F.S., Harris, P.C., King, B.F., Torres, V.E., Erickson, B.J., 2017. Performance of an artificial multi-observer deep neural network for fully automated segmentation of polycystic kidneys. Journal of Digital Imaging 30, 442–448.

[26] Koohbanani, N.A., Jahanifar, M., Tajadin, N.Z., Rajpoot, N.M., 2020. Nuclick: A deep learning framework for interactive segmentation of microscopy images. Medical image analysis 65.

[27] Kusakunniran, W., Saiviroonporn, P., Siriapisith, T., Tongdee, T., Uraiverotchanakorn, A., Leesakul, S., Thongnarintr, P., Kuama, A., Yodprom, P., 2023. Automatic measurement of cardiothoracic ratio in chest x-ray images with progan-generated dataset. Applied Computing and Informatics URL: https://doi.org/10.1108/ACI-11-2022-0322, doi:10.1108/ACI-11-2022-0322.

[28] Le'Clerc Arrastia, J., Heilenkötter, N., Otero Baguer, D., Hauberg-Lotte, L., Boskamp, T., Hetzer, S., Duschner, N., Schaller, J., Maass, P., 2021. Deeply Supervised UNet for Semantic Segmentation to Assist Dermatopathological Assessment of Basal Cell Carcinoma. J Imaging 7.

[29] Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H., 2019. Expectation-maximization attention networks for semantic segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) , 9166–9175.

[30] Liao, X., Li, W., Xu, Q., Wang, X., Jin, B., Zhang, X., Zhang, Y., Wang, Y., 2020. Iteratively-refined interactive 3d medical image segmentation with multi-agent reinforcement learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 9391–9399.

[31] Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Dollár, P., 2017. Focal loss for dense object detection. 2017 IEEE International Conference on Computer Vision (ICCV) , 2999–3007.

[32] Lin, Z., Duan, Z.P., Zhang, Z., Guo, C.L., Cheng, M.M., 2022. Focuscut: Diving into a focus view in interactive segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2637–2646.

[33] Mikhailov, I., Chauveau, B., Bourdel, N., Bartoli, A., 2022. A deep learning-based interactive medical image segmentation framework, in: Wu, S., Shabestari, B., Xing, L. (Eds.), Applications of Medical Artificial Intelligence, Springer Nature Switzerland, Cham. pp. 98–107.

[34] Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D., 2021. Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence , 1–1.

[35] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J., 2020. Deep learning vs. traditional computer vision, in: Arai, K., Kapoor, S. (Eds.), Advances in Computer Vision, Springer International Publishing, Cham. pp. 128–144.

[36] Ramadan, H., Lachqar, C., Tairi, H., 2020. A survey of recent interactive image segmentation methods. Computational Visual Media 6, 355 – 384.

[37] Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and

Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham. pp. 234–241.

[38] Rother, C., Kolmogorov, V., Blake, A., 2004. "grabcut": Interactive foreground extraction using iterated graph cuts, in: ACM SIGGRAPH 2004 Papers, Association for Computing Machinery, New York, NY, USA. p. 309–314. doi:10.1145/1186562.1015720.

[39] Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536. doi:10.1038/323533a0.

[40] Sakinis, T., Milletari, F., Roth, H.R., Korfiatis, P., Kostandy, P.M., Philbrick, K.A., Akkus, Z., Xu, Z., Xu, D., Erickson, B.J., 2019. Interactive segmentation of medical images through fully convolutional neural networks. ArXiv abs/1903.08205.

[41] Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., Xue, Z., Shen, D., Shi, Y., 2020. Lung infection quantification of covid-19 in ct images with deep learning. ArXiv .

[42] Shvets, A.A., Iglovikov, V.I., Rakhlin, A., Kalinin, A.A., 2018. Angiodysplasia detection and localization using deep convolutional neural networks. bioRxiv .

[43] Siddique, N., Paheding, S., Elkin, C.P., Devabhaktuni, V., 2021. U-net and its variants for medical image segmentation: A review of theory and applications. IEEE Access 9, 82031–82057.

[44] Siddiquee, M.M.R., Myronenko, A., 2021. Redundancy reduction in semantic segmentation of 3d brain tumor mris. ArXiv abs/2111.00742.

[45] Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G.J.S., Menze, B.H., Ronneberger, O., Summers, R.M., Bilic, P., Christ, P.F., Do, R.K.G., Gollub, M.J., Golia-Pernicka, J., Heckers, S., Jarnagin, W.R., McHugo, M., Napel, S., Vorontsov, E., Maier-Hein, L., Cardoso, M.J., 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. ArXiv abs/1902.09063.

[46] Sofiiuk, K., Petrov, I.A., Barinova, O., Konushin, A., 2020. F-brs: Rethinking backpropagating refinement for interactive segmentation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 8620–8629.

[47] Sofiiuk, K., Petrov, I.A., Konushin, A., 2021. Reviving iterative training with mask guidance for interactive segmentation. 2022 IEEE International Conference on Image Processing (ICIP) , 3141–3145.

[48] Tustison, N.J., Avants, B.B., Cook, P.A., Zheng, Y., Egan, A., Yushkevich, P.A., Gee, J.C., 2010. N4itk: Improved n3 bias correction. IEEE Transactions on Medical Imaging 29, 1310–1320.

[49] Wang, G., Li, W., Zuluaga, M.A., Pratt, R., Patel, P.A., Aertsen, M., Doel, T., David, A.L., Deprest, J.A., Ourselin, S., Vercauteren, T.K.M., 2018. Interactive medical image segmentation using deep learning with image-specific fine tuning. IEEE Transactions on Medical Imaging 37, 1562–1573.

[50] Wang, G., Zuluaga, M.A., Li, W., Pratt, R., Patel, P.A., Aertsen, M., Doel, T., David, A.L., Deprest, J.A., Ourselin, S., Vercauteren, T.K.M., 2019. Deepigeos: A deep interactive geodesic framework for medical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 1559–1572.

[51] Xie, Q., Hovy, E.H., Luong, M.T., Le, Q.V., 2020. Self-training with noisy student improves imagenet classification. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 10684–10695.

[52] Zhou, B., Chen, L., Wang, Z., 2019. Interactive deep editing framework for medical image segmentation, in: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.T., Khan, A. (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2019, Springer International Publishing, Cham. pp. 329–337.

[53] Zhou, T., Li, L., Bredell, G., Li, J., Konukoglu, E., 2022. Volumetric memory network for interactive medical image segmentation. Medical image analysis 83.

[54] Zhu, H., Meng, F., Cai, J., Lu, S., 2016. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. Journal of Visual Communication and Image Representation 34, 12–27.

[55] Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.V., 2020. Rethinking pre-training and self-training. ArXiv abs/2006.06882.