SkipClick: Combining Quick Responses and Low-Level Features for Interactive Segmentation in Winter Sports Contexts

Robin Schön   Julian Lorenz   Daniel Kienzle

Rainer Lienhart

University of Augsburg

Germany, 86159 Augsburg, Universitätsstr. 6a

{robin.schoen, julian.lorenz, daniel.kienzle, rainer.lienhart}@uni-a.de

Abstract

In this paper, we present a novel architecture for interactive segmentation in winter sports contexts. The field of interactive segmentation deals with the prediction of high-quality segmentation masks by informing the network about the objects position with the help of user guidance. In our case the guidance consists of click prompts. For this task, we first present a baseline architecture which is specifically geared towards quickly responding after each click. Afterwards, we motivate and describe a number of architectural modifications which improve the performance when tasked with segmenting winter sports equipment on the WSESeg dataset. With regards to the average NoC@85 metric on the WSESeg classes, we outperform SAM and HQ-SAM by 2.336 and 7.946 clicks, respectively. When applied to the HQSeg-44k dataset, our system delivers state-of-the-art results with a NoC@90 of 6.00 and NoC@95 of 9.89. In addition to that, we test our model on a novel dataset containing masks for humans during skiing. [1]

1. Introduction

One of the most important applications of computer vision to sports, and therefore to winter sports as well, is the exact localization of athletes and objects involved in the activity. A large part of the attention falls upon the localization of the athlete. This mostly happens in the form of pose estimation [33] and athlete segmentation [3, 24] . Recently, however, there has been an increased focus on the localization of the equipment used in the sport activities. In some cases this takes the form of segmenting the sports equipment [16, 40] while other papers make use of the segmentation mask for various additional purposes [31, 32]. This development entails an increased need for annotated data in
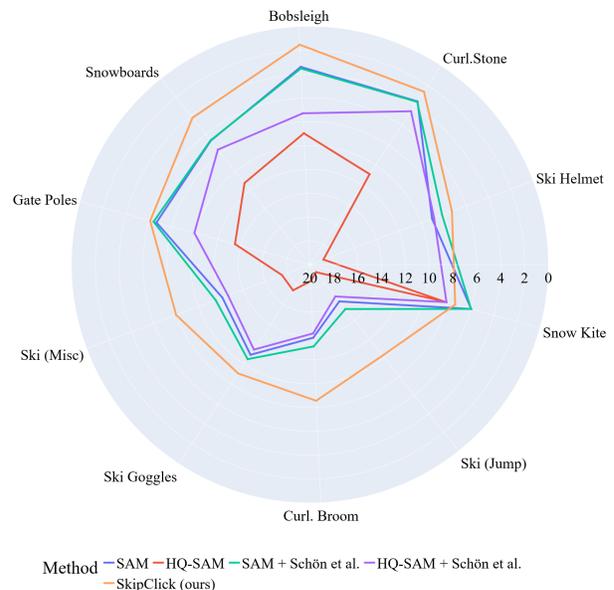

Figure 1. A comparison of the performance of SkipClick with other methods on the WSESeg [42] dataset. The metric is NoC@85.

this direction. On top of this, segmentation masks take a lot of time to annotate. Especially with regards to annotation tools that allow the user to annotate polygons or use brush / eraser tools, annotating fine details can be very challenging when annotation speed also constitutes an important factor. The COCO dataset [26] for example is subject to slight errors in the annotated segmentation masks. For this reason, interactive segmentation systems have been developed. The purpose of these systems lies in aiding the user with segmenting an image by trying to infer high quality image segmentation masks from used guidance. The main goal is to enable the user to produce such desired object segmentation masks in much less time than would be necessary to

---

[1] We will release code and the dataset upon publication: https://github.com/Schorob/skipclick

annotate manually from scratch. In most cases this amounts to clicks and scribbles to indicate the foreground / background on the images, or bounding boxes around the object that the user wants to segment. In recent years, neural network models have become the main component of such interactive segmentation systems. In these cases the network is given the image, the user guidance and in some cases a coarse mask of yet insufficient quality. The network is then tasked with predicting a segmentation mask for a desired object on the image, as indicated by the user guidance.

In this paper we will only focus on iterative mask refinement with the help of foreground and background clicks: The user repeatedly inspects a mask and places a click on an erroneously segmented area. The segmentation network then uses the image, previous mask and the user clicks to predict an improved mask. This process is repeated until the user considers the object to be segmented adequately.

Most earlier frameworks feed the clicks and previous mask to the model in conjunction with the image. In this situation which we call *early fusion*, the early layers of the segmentation model have access to the prompts. Whilst generally allowing for improved mask quality, this entails a drawback with respect to the response time after each click: The entire model has to be run again in order to generate a new mask. One of the most notable developments is the Segment Anything Model [20]. In order to increase the interaction speed, SAM first computes a prompt-independent image feature tensor. This features tensor is then fused with the prompt information by two lightweight networks: a prompt encoder and a mask decoder. Whenever the user adds a new click, only the two lightweight networks have to be rerun, allowing for a quick response of the system. We refer to this as *late fusion*. Since SAM is trained on the SA-1B dataset, which contains 1.1 billion masks, it constitutes a viable foundation model. The authors of [42], however, mention a crucial problem of SAM. Despite being trained on the largest dataset for interactive segmentation to date, it does not generalize well to the domain of winter sports equipment. The authors demonstrate this on their new WS-ESeg dataset.

In this paper we will introduce a **new architecture for interactive segmentation in unusual domains** such as winter sports equipment, which we will call *SkipClick*. The first and most important requirement of our model will be a **quick response time**. Secondly, we have to make sure to **avoid overfitting** since snow landscapes are a rather unusual domain in comparison to the regular consumer photos that constitute most available datasets. On top of this, we have to **explicitly enable the network to deal with fine structures** that sometimes occur when segmenting winter sports equipment. In addition to constructing a model that adheres to these requirements, we will be **able to demonstrate that our model is not over-engineered to the win-ter sports domain**. When compared to other real-time interactive segmentation models, our model shows competitive performance on datasets that contain only regular consumer photos. Since [42] focuses on winter sports equipment instead of the athletes themselves, **we propose a new dataset with masks for skiers**. This new dataset is called SHSeg (Skiing Human Segmentation) and contains 534 segmentation masks on 496 images. This new dataset will be used as an evaluation dataset.

Our contributions can be summarized as follows:

- We present a real-time interactive segmentation model which is capable of performing segmentation on unusual domains such as winter sports. When designing the architecture, we pay explicit attention to enable the model to deal with fine-grained structures in the images.

- We provide an ablation study of our model on the winter sports equipment segmentation dataset WSESeg. We are able to show that our model even surpasses the performance of SAM, despite being trained on the vastly smaller COCO+LVIS dataset.

- We are also able to show that our model is not over-engineered to the winter sports domain. The model exhibits competitive or even superior performance on general consumer image datasets when compared to other late fusion based models.

- We present a novel segmentation dataset with masks for athletes during skiing.

2. Related Work

2.1. Usage of Segmentation Masks in Sports

In [3], the authors propose the segmentation of participants of soccer and basketball matches. The authors of [9] present a system for segmenting players and localizing the ball conjointly, while [8] focuses on the player localization aspect. The authors of [14] also segment people in sports contexts, but include the used equipment in their masks. The method proposed in [45] aims at directly segmenting the basketball field itself. Similarly, the authors of [30] segment the fields in outdoor sports. [40] track the sword in fencing matches by localizing the tip and segmenting the sword. The authors of [16] estimate the surface of basketballs by estimating their location and radius. The authors of [31, 32] use the segmentation masks of skis to enable the detection of additional keypoints. [42] is the most relevant publication to our work. Therein, the authors publish a novel dataset for the segmentation of winter sports equipment and techniques for the online adaptation of interactive segmentation methods.

2.2. Interactive Segmentation

Interactive segmentation deals with problem of producing high quality image segmentation masks by integrating information from user interactions. While there are certain methods which make use of hand-crafted image features and pixel values [1, 10, 12, 39], most modern methods employ neural networks to perform this task. The most prevalent form of interaction are clicks that indicate single foreground and background pixels. The authors of [44] discuss various different networks and the effectiveness of using the previous step's mask as input. The work in [28] introduces transformers as an effective choice of backbone for interactive segmentation. The methods presented in [17, 43] use backpropagation to enforce consistency between the user interaction and the mask outputs. Most of the aforementioned methods provide masks of great quality at the cost of a slow response after each click. The authors of [20] introduce the concept of late fusion, provide the large SA-1B dataset and publish the weights to be used as a foundation model. [18] introduces the HQSeg-44k dataset and fine tunes SAM to increase the level of detail in the predicted masks. [15] provides another architecture using late fusion to achieve an efficient response time. The baseline we arrive at in Sec. 3.2 resembles [27] to a certain degree. There also has been an effort to adapt interactive segmentation systems during usage, as can be seen in [22, 23, 41, 42].

3. Method

We will first describe the problem of interactive segmentation by presenting how the task is usually structured. Afterwards we will describe our architecture in two stages. In the first stage, we will present a baseline based on certain requirements: A quick response time after the clicks, a shallow decoder and a backbone with a strong capacity for generalization. This baseline can be found in Fig. 2a. In the second stage, we present certain architectural modifications and show that they improve the performance of the model. Due to a lack of large scale winter sports related segmentation masks, we want the model be able to generalize well. Since we aim to segment winter sports equipment, which often has a surface with delicate structures, we pay special attention to enable our model to segment fine details by using low-level image feature tensors. The resulting architecture can be found in Fig. 2b. Table 1 displays the measured improvements.

3.1. Problem Statement

In interactive segmentation, we have an image $\mathbf{x}_{img} \in \mathbb{R}^{H \times W \times 3}$ which contains an object of interest. We want to create a high-quality segmentation mask $\mathbf{m} \in \{0, 1\}^{H \times W}$ which has the value 1 for all pixels belonging to our object of interest and the value 0 for all other pixels. We thus ef-

fectively want to distinguish foreground from background. In order to achieve this, the user repeatedly interacts with a segmentation network $\mathcal{F}_{ISeg}$. Each of these interactions has the following structure:

1. During the τ-th interaction the user inspects the current mask $\mathbf{m}_\tau$ for erroneous areas. In the beginning of the first iteration the initial mask $\mathbf{m}_0$ only consists of 0s (all background).

2. In order to give the network additional information, the user places a click $\mathbf{p}_\tau = (i_\tau, j_\tau, l_\tau)$ on the image. Here, $(i_\tau, j_\tau) \in \{1, ..., H\} \times \{1, ..., W\}$ is an image coordinate. Whether the clicked pixel belongs to the foreground (+) or background (-) is indicated by the label $l_\tau \in \{+, -\}$. The user sets the label by choosing to place the click either with the left or right mouse button.

3. The image $\mathbf{x}_{img}$, the current mask $\mathbf{m}_\tau$, and all accumulated clicks $\mathbf{p}_{0:\tau}$ so far are given to the network in order to produce an improved mask $\mathbf{m}_{\tau+1} = \mathcal{F}_{ISeg}(\mathbf{x}_{img}, \mathbf{m}_\tau, \mathbf{p}_{0:\tau})$.

Since the user gives an increasing amount of information to the network, we assume this method to offer the possibility to eventually arrive at a mask of very high quality. A good interactive segmentation system is thus characterized by minimizing the number of clicks necessary to create such a high quality mask.

3.2. Architecture

We will first describe a baseline architecture, which is held as simple as possible while still fulfilling a certain range of requirements. By doing so, we arrive at an architecture that bears some similarity to [27]. Their method also uses attention to integrate prompts, a ViT encoder and the mask predictor follows [28]. However, in contrast to [27], we use full transformer encoder blocks to fuse the features and refrain from using cross attention. After introducing the baseline, we look at some modifications and justify each of them. This will eventually lead to our complete SkipClick architecture.

The baseline. We want to offer a simple baseline architecture, which generalizes well and equips the network with the capability to deliver real-time responses after each click. As a consequence we aim for the module which combines the interaction information and the image features to remain relatively shallow. Apart from the mask predictor, where we follow common practice, our baseline will be an almost direct consequence of the aforementioned requirements.

The first requirement is a large potential for generalization, since we want to be able to segment a vast variety

(a) The baseline architecture.
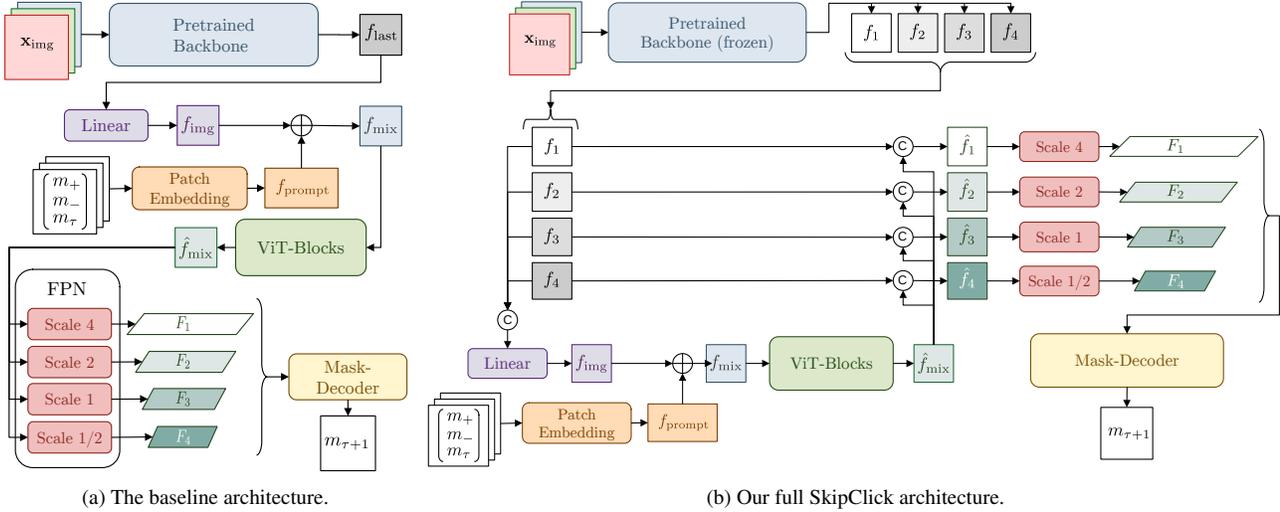
(b) Our full SkipClick architecture.

Figure 2. The baseline (*left*) and the SkipClick (*right*) architecture. Note that the bulk of the computation happens in the backbone, which only has be executed once per image. Freezing the backbone during training enables the backbone to retain its generality from unsupervised pretraining. The use of multi-level features and skip connections allows the model to deal with fine structures encountered when segmenting winter sports equipment.

of shapes in various different contexts. Here we will follow common practice and use a vision transformer architecture [20, 28]. Interactive segmentation is generally class agnostic in nature. Thus, we do not want our backbone to be constrained by the bias induced when pre-training on a labeled dataset. We will opt for a ViT [4] that has been pretrained with the DINOv2 framework [35], since this pretext task does not make use of labels. In our baseline, the input image is transformed into an image feature tensor

$$f_{\text{img}} = \text{Linear}(\text{ViTBackbone}(\mathbf{x}_{\text{img}})) \in \mathbb{R}^{\frac{H}{14} \times \frac{W}{14} \times d_{\text{model}}}, \tag{1}$$

where H, W and d_{model} are the image height, image width and the feature dimension in the transformer, respectively. The second requirement consists in the desire to get a real-time response from the neural network. Until recently, interactive segmentation methods reran the entire network after each click [2, 28, 44]. Therein, the input image was fused with the previous mask and the foreground / background clicks very early in the network. This had the side effect of having to repeat the bulk of the computation after every interaction. While these types of network usually perform really well with respect to the number of clicks, the slow response time is a hindrance to the user during actual usage. Recent publications, such as SAM [20] and Inter-Former [15], have tackled this problem by splitting the network into a heavyweight image encoder and a lightweight mask predictor. The heavyweight image encoder constitutes the bulk of the networks computation and only needs to be run once, since the image itself never changes during the segmentation interactions. After each interaction, the

lightweight mask predictor is given the image features, the previous mask and the encoded clicks in order to predict an improved version of the mask. Since only the lightweight mask predictor has to be executed after each interaction, the response time the user experiences diminishes drastically. We will adopt this strategy of only encoding the image once, and executing a lightweight network after each interaction.

We combine the image features f_{img} with the user guidance as follows: We encode the positive and negative clicks in $\mathbf{p}_{0:\tau}$ by drawing small disks with a radius of 5 pixels on two different binary images. This results in two click maps $\mathbf{m}_+, \mathbf{m}_- \in \{0, 1\}^{H \times W}$. These click maps are then concatenated with the mask from the previous round $\mathbf{m}_\tau$ and fed to a patch embedding layer to obtain

$$f_{\text{prompt}} = \text{PatchEmbedding}(\text{Concat}(\mathbf{m}_+, \mathbf{m}_-, \mathbf{m}_\tau)), \tag{2}$$

where the patch size is 14×14 as in DINOv2. In order to combine the image features and the prompt features, we add the two tensors in an element-wise fashion to obtain

$$f_{\text{mix}} = f_{\text{img}} + f_{\text{prompt}}. \tag{3}$$

Since we aim for a lightweight architecture, we will need it to be relatively shallow. Because we want to guarantee for a sufficiently large field of view, whilst stacking the models in a shallow fashion, we use transformer blocks to incorporate the image features and the interaction information. This results in

$$\hat{f}_{\text{mix}} = \text{ViTBlocks}(f_{\text{mix}}). \tag{4}$$

When it comes to the mask head, we will use the same head as SimpleClick: First the feature tensor is fed to an feature

pyramid network (FPN) as described in [25]. This feature pyramid is then given to a SegFormer decoder [46] to obtain

$$\mathbf{m}_{\tau+1} = \text{MaskDecoder}(\text{FPN}(\hat{f}_{\text{mix}})). \quad (5)$$

It should be noted that our baseline is similar to the method proposed in [27], with two main differences. First, we use transformer blocks and not just the attention mechanism to incorporate interactions and image features. Second, we do not use any form of cross-attention or text integration.

SkipClick. We will use our central ablation study to motivate the decisions in constructing our architecture in a step-by-step fashion. While doing so, we will show the improvements incurred by each design decision. Therefore, we recommend any reader unacquainted with interactive segmentation and the WSESeg dataset to first read Section 4.1. The full architecture can be seen in Fig. 2b and the improvements incurred by the architectural elements in Tab. 1.

The most important aspect of dealing with winter sports in machine learning settings is generalization. Especially winter sports equipment occurs rather scarcely in datasets made up of general consumer images. Although snowboards and skis happen to be part of the COCO dataset, their masks are often only coarse polygons of mediocre quality. Since the segmentation training data is implicitly limited to certain object types, we incur a risk of overfitting. Especially the backbone might loose some of its generality. In order to alleviate this problem take a step that may seem rather counterintuitive. We freeze the backbone which is used to encode the image. Our results corroborate this decision: When comparing the first two lines in Tab. 1, we see an improvement of the average NoC@85 metric (defined in Section 4.1) from 9.463 to 9.416.

Recent techniques [20,27,28] rely on the last feature tensor computed by the ViT encoder to still contain enough fine-grained information about the image to perform pixel-level tasks such as segmentation. Especially in the winter sports domain, where some objects may have delicate details in their shapes, such fine-grained information is essential when producing novel segmentation masks. We assume the information from the features generated by the last layer to lack low-level details. We therefore use various intermediate feature tensors as well. In particular we use the feature tensors after the 3rd, 6th, 9th and 12th encoder block of the ViT-B, which we call f_1, f_2, f_3 and f_4, respectively.

$$f_1, f_2, f_3, f_4 = \text{ViTBackbone}(\mathbf{x}_{\text{img}}) \quad (6)$$

We integrate the information from various layers in the backbone by concatenating their output tensor and feeding them to a linear layer. This results in an alteration of Eq. (1) to compute

$$f_{\text{img}} = \text{Linear}(\text{Concat}(f_1, f_2, f_3, f_4)). \quad (7)$$

This change incurs a considerable improvement, lowering NoC@85 from 9.416 to 7.285, whilst only extending the baseline by a single linear layer.

So far our network only has access to the fine-grained information *before* the integration of the prompts happens. We do however want our network to be able to access the intermediate features tensors *after* the prompts are integrated, since the prompts are necessary for the network to know what object we want to segment at all. Inspired by U-Net [37], we choose to integrate skip connections (see Fig. 2b). This helps us in creating the intermediate tensors

$$\hat{f}_i = \text{Concat}(\hat{f}_{\text{mix}}, f_i) \qquad \text{for } i = 1, 2, 3, 4. \quad (8)$$

Right now, we have $\hat{f}_i \in \mathbb{R}^{\frac{H}{14} \times \frac{W}{14} \times 2 \cdot d_{\text{model}}}$. In order to obtain a feature pyramid as in [25] the tensors are rescaled to $4\times, 2\times, 1\times$ and $\frac{1}{2}\times$ their size as show in Fig. 2b. The rescaling modules do not simply upsample/downsample, but instead use (transposed) convolutions and non-linearities as in [28]. The resulting feature maps F_1, F_2, F_3 and F_4 are processed by a SegFormer decoder to obtain

$$\mathbf{m}_{\tau+1} = \text{MaskDecoder}(F_1, F_2, F_3, F_4). \quad (9)$$

Bridging low-level features to the feature pyramid network improves our model again, reducing the NoC@85 from 7.285 to 6.494. When looking at all the architectural changes in Tab. 1 in total, we observe a drop of 2.969 in the NoC@85 metric and a drop of 2.868 in the NoC@90 metric.

4. Experiments

4.1. Experimental Setting

Metric. As mentioned in section 3.1, the main indicator of the quality of an interactive segmentation system is number of clicks necessary to create a high quality mask. We will test our system on datasets with pre-existing ground truth masks $\mathbf{m}_{\text{GT}}$. First, we fix an IoU threshold θ_{IoU}. A predicted mask $\mathbf{m}_\tau$ will be considered of sufficient quality if its IoU with $\mathbf{m}_{\text{GT}}$ reaches this threshold as in

$$\text{IoU}(\mathbf{m}_\tau, \mathbf{m}_{\text{GT}}) = \frac{|\mathbf{m}_\tau \cap \mathbf{m}_{\text{GT}}|}{|\mathbf{m}_\tau \cup \mathbf{m}_{\text{GT}}|} \geq \theta_{\text{IoU}}. \quad (10)$$

The number of clicks (NoC) necessary to create such a sufficient mask is the most important metric in interactive segmentation. In some cases the system is effectively incapable to segment the mask, which might incur a potentially unbounded number of simulated interactions. To account for this problem, the maximum number of interactions is capped to 20. The resulting metric is called $\text{NoC}_{20}@\theta_{\text{IoU}}$. Most notably, a lower NoC metric value indicates a better performance. We follow [44] for the simulation of user clicks. A detailed description of the clicks sampling algorithm can be found in the supplementary material.

Configuration	Bobsleigh		Curl. Stone		Ski Helmet		Snow Kite		Ski (Jump)	
	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
Baseline	4.48	6.15	6.53	8.29	7.95	10.61	9.1	10.82	15.05	18.26
+ Frozen Backbone	4.01	5.64	6.11	7.98	7.55	10.54	9.32	10.77	15.02	17.96
+ Intermediate Features	1.6	2.19	3.21	4.99	8.22	10.82	7.5	9.56	11.5	16.43
+ Skip Connections	1.52	2.08	2.61	4.09	7.27	9.31	7.36	8.93	10.23	15.51
Curl. Broom		Ski Goggles		Ski (Misc)		Slalom Gate Poles		Snowboards		**Average**	
NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
13.55	16.42	9.14	12.38	12.17	15.03	9.4	13.12	7.26	9.23	9.463	12.031
14.02	16.62	8.9	12.33	12.36	15.11	9.9	13.65	6.97	8.91	9.416	11.951
10.63	15.12	8.97	12.38	9.23	13.02	7.24	12.19	4.75	6.74	7.285	10.344
8.56	13.16	9.06	11.5	8.01	11.49	6.1	9.62	4.22	5.94	**6.494**	**9.163**

Table 1. Results on the WSESeg dataset validating our architectural choices. The rows describe our architecture in a cumulative manner, meaning each row also contains the configurational aspects of the rows above. The final configuration is our SkipClick architecture. The lower the NoC metric, the better the performance.

Configuration	Bobsleigh		Curl. Stone		Ski Helmet		Snow Kite		Ski (Jump)	
	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
SAM [20]	3.38	5.33	3.59	6.26	9.05	12.70	5.97	8.23	16.05	18.43
HQ-SAM [18]	8.96	11.83	10.90	13.44	18.79	19.58	8.34	11.04	19.19	19.88
SAM + Schön et al. [42]	3.51	5.45	3.64	6.39	8.15	12.41	5.98	8.21	15.23	18.42
HQ-SAM + Schön et al. [42] †	7.28	8.24	4.58	7.71	8.87	12.85	8.12	10.42	16.58	18.98
SkipClick (Ours)	1.52	2.08	2.61	4.09	7.27	9.31	7.36	8.93	10.23	15.51
Curl. Broom		Ski Goggles		Ski (Misc)		Slalom Gate Poles		Snowboards		**Average**	
NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
13.86	17.73	10.94	14.57	12.15	15.41	6.65	10.46	6.68	9.49	8.83	11.86
18.61	19.68	17.42	18.24	17.50	18.74	13.48	17.12	11.22	13.59	14.44	16.31
13.13	17.85	10.50	14.41	11.55	15.49	6.40	10.46	6.65	9.45	8.48	11.85
14.22	17.56	11.46	14.82	12.66	16.67	9.95	14.18	7.64	10.48	10.14	13.19
8.56	13.16	9.06	11.5	8.01	11.49	6.1	9.62	4.22	5.94	**6.494**	**9.163**

Table 2. A comparison of SkipClick with previous methods on the WSESeg dataset. Most of the numerical results are taken from [42]. †: When combining HQ-SAM with the method from [42], one of their ablations performs vastly better than their full method. For this reason we used the NoC metrics from their better performing ablation in this table row.

Method	NoC@85	NoC@90
SAM [20]	3.95	7.46
HQ-SAM [18]	10.60	14.29
SAM + Schön et al. [42]	3.88	7.47
HQ-SAM + Schön et al. [42] †	4.70	8.45
SkipClick (Ours)	**1.44**	**2.52**

Table 3. A comparison of SkipClick with previous methods on our newly proposed SHSeg dataset. †: Instead of taking the full method, we take the ablation that delivered the best performance on WSESeg in [42], as we did in Tab. 2.

Implementation Details. In all of our experiments we use a Adam optimizer [19] with a learning rate of $\mu = 5 \cdot 10^{-5}$ and $\beta_1 = 0.9, \beta_2 = 0.999$. For all experiments we use focal loss [38]. We train on a combination of COCO and LVIS [11,26] for 55 epochs. It should be noted that one epoch is defined as 30,000 images. We downscale our learning rate by $\frac{1}{10}$ after epoch 50 and 55. We follow the practice of [28] and initially sample up to 24 random points. Iterative training with simulated points is carried out for three additional iterations. Our backbone is a ViT-B [4] that has been pretrained with DINOv2 [35]. During training we use random crops of size 448×448. We also rescale our im-

ages to this during testing unless said otherwise. For the WSESeg, SHSeg and HQSeg-44k datasets we use a resolution of 896×896 in order be able compare our system to SAM and HQ-SAM, which use a resolution of 1024×1024. For the DAVIS dataset we use a resolution of 672×672. All our resolutions stem from the requirement of being multiple of the patch size 14×14.

Datasets. We follow common practice and use the COCO+LVIS dataset for training. This dataset constitutes a combination of the COCO dataset [26] along with its annotations, and the LVIS dataset [11] that has been introduced in [44]. Our main ablation study is carried out on the WSESeg dataset for the segmentation of winter sports equipment. This dataset contains 7452 masks for ten different classes of objects. We also carry out some evaluations on datasets which are commonly used in interactive segmentation: GrabCut [39], Berkeley [34], DAVIS [36] and SBD [13]. In addition to this, we make use of the HQSeg-44k dataset [11], which effectively is a mix of various high-quality human-annotated datasets. Especially DAVIS and HQSeg-44k allow us to test the efficacy of the architectural

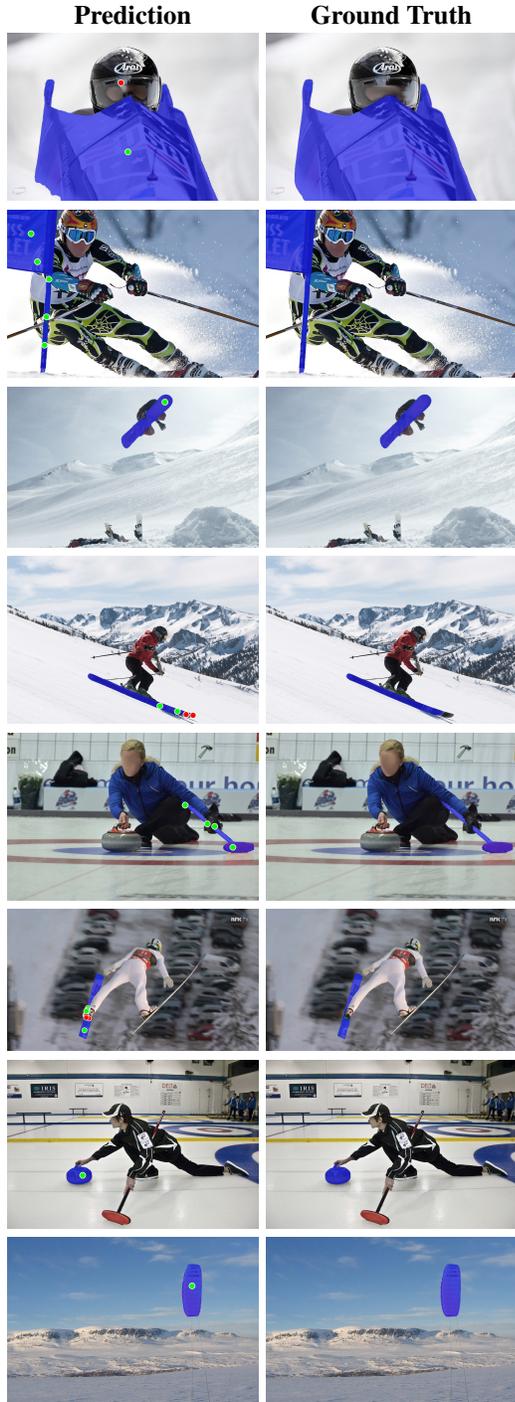|	Prediction	Ground Truth

Figure 3. Qualitative examples on WSESeg. Foreground clicks are *green*, background clicks are *red* and the masks are *blue*.

modifications on data which contains fine-grained details.

The SHSeg dataset. We present a novel dataset containing masks for skiing humans, which we call *SHSeg* for

Configuration	GrabCut NoC@90	Berkeley NoC@90
Baseline	1.74	3.86
+ Frozen Backbone	1.72	3.73
+ Intermediate Features	**1.40**	2.84
+ Skip Connections	1.44	**2.45**
DAVIS NoC@90	SBD NoC@90	HQSeg-44k NoC@90
6.75	8.53	7.91
6.40	8.51	7.98
5.05	6.65	6.40
4.94	**6.18**	**6.00**

Table 4. Ablation study on general consumer images. A lower NoC metric indicates better performance.

Skiing Human Segmentation. Since WSESeg mostly focuses on the segmentation of the equipment in winter sports contexts, this data allows to test our models capacity to segment the athletes themselves. The images themselves originate from the SkiTB dataset [5,6], and are already annotated with one bounding box of the skier per image. In order to select the images we use, we first filter out all images whose corresponding bounding box has a height or width of less than 150 pixels. Out of the remaining images, we randomly sample 500 images. We only annotate images that show a clearly distinguishable skier, which leaves us with 496 images. Since some images contain more than one athlete, we end up with 534 masks in total. We annotated the dataset with the system provided in [28], since using our own model would have introduced a bias. It should be noted that, while the annotations are created using interactive segmentation systems, all annotations are inspected and corrective clicks are provided by a human until a very high degree of quality is reached.

4.2. Results

The design aspects of our architecture have already been validated for winter sports in Tab. 1 in Sec. 3.2. In Tabs. 2 and 3 we compare ourselves with SAM [20], HQ-SAM [18] and the method by [42] which has been applied to the two architectures. The SAM (Segment Anything Model) architecture introduced the strategy of late fusion in interactive segmentation and has been trained on the largest interactive segmentation dataset to date: The SA-1B dataset, which has been published alongside the model and contains 1.1B masks on 11 million images. Despite such an extensive amount of training data, we observe an inferior performance when compared to our SkipClick model, which has been trained on the vastly smaller COCO+LVIS dataset (99k images with 1.5 million masks). SAM achieves a NoC@85 of 8.83 in comparison to our SkipClick model with a NoC@85 of 6.49. The slightly more challenging NoC@90 metric supports this observation with 11.86 for SAM and 9.16 for

Method	Fusion type	DAVIS			HQSeg-44K		
		NoC@90	NoC@95	≥ 20@95	NoC@90	NoC@95	≥ 20@95
RITM [44]	*early*	5.34	11.45	139	10.01	14.85	910
FocalClick [2]	*early*	**4.90**	10.40	123	7.03	10.74	649
SimpleClick [28]	*early*	5.06	10.37	**107**	7.47	12.39	797
SAM [20]	*late*	5.14	10.74	154	7.46	12.42	811
MobileSAM [47]	*late*	5.83	12.74	196	8.70	13.83	951
HQ-SAM [18]	*late*	5.26	**10.00**	136	6.49	10.79	671
SegNext [27]	*late*	5.34	12.80	163	7.18	11.52	700
InterFormer [15]	*late*	5.45	11.88	150	7.17	10.77	658
SkipClick (ours)	*late*	4.94	11.92	158	**6.00**	**9.89**	**608**

Table 5. Segmentation of images with fine granularity in their segmentation masks. Our SkipClick method achieves state-of-the-art results on HQSeg-44k, which corroborates the general applicability of our method. ≥ 20@95 denotes the number of masks for which the test run needed at least 20 clicks to attain a IoU of at least 95 with the pre-existing ground truth. For all metrics, lower is better. Results for other methods are from [27].

SkipClick. We mainly attribute the performance improvement to the fact that our model makes extensive use of low-level features, while SAM only uses the tensor produced by the last layer of its image encoder. The latter hypothesis is especially supported by looking at our baseline in Tab. 1, which does not use intermediate features and only achieves an average NoC@85 of 9.46. The baseline is thus worse than SAM, while the usage of intermediate features results in a model that is better than SAM. Our model outperforms HQ-SAM on all classes. Even when accounting for the improvements incurred by the method in [42], on average, our model still outperforms both SAM and HQ-SAM. Figure 1 displays our comparison on WSESeg as a radar plot. Examples for qualitative results can be found in Fig. 3. The results on our novel SHSeg dataset corroborate our comparison. Our SkipClick model reduces the NoC@85 metric by 62.8 % in comparsion to the second best method (SAM + Schön et al.) and the NoC@90 metric by 66.2 % in comparison to SAM. Qualitative examples on the SHSeg dataset can be found in the supplementary material. We also measured the speed of the methods in Tab. 2 in terms of the time the model needs to respond after each click. The slowest methods are the modifications of SAM and HQ-SAM provided by Schön et al. [42], which have a response time of 41.38 ms and 56.13 ms, respectively. This is caused by the necessity of an additional backpropagation in their framework. The standard SAM and HQ-SAM need 15.01 ms and 18.83 ms to respond. Out of the compared methods, ours is the fastest with a response time of 6.61 ms. All these durations have been measured using a V100 GPU.

In Tab. 4, we also evaluate our main ablation study on datasets that comport with general consumer images. The results on these datasets generally support the results obtained during the method construction (seen in Tab. 1). In almost all cases our complete methods delivers the strongest results except for one: The skip connections slightly worsen the performance on GrabCut. We assume this to be the case due to a slight overfitting to particularities of COCO+LVIS in comparison to the GrabCut dataset.

We also compare our method to existing methods on the DAVIS and HQSeg-44k datasets in Tab. 5. Both datasets are characterized by a high level of detail in their masks. With respect to the NoC@90 metric on DAVIS, our model performs best when compared to the other late fusion models with 4.94 clicks on average. If we also compare to the early fusion models, we are only second best with FocalClick having a NoC@90 of 4.90. On HQSeg-44k our model achieves state-of-the-art results for all explored metrics. The results in Tabs. 4 and 5 demonstrate that our architectural design has not been overfit to the winter sports domain, but instead performs also well on other domains.

5. Conclusion

In our paper we presented a new architecture for interactive segmentation on the winter sports domain. We first describe a baseline where we pay particular attention to equip the model with the capacity to deliver real-time responses after each clicks, by employing late fusion. Afterwards, we add certain architectural features for which we are able to show that they improve the performance when tasked with segmenting winter sports equipment. We are able to show that our model outperforms SAM and HQ-SAM on the WSESeg dataset. We manage to confirm this result on a newly published dataset for the segmentation of skiers, called SHSeg. Additionally, when carrying out our ablation study on general consumer image datasets, we are able to show that our model's architecture is not overfit to the domain of winter sports equipment. When comparing our method to existing methods, we achieve competitive results on the DAVIS dataset and state-of-the-art results on the HQSeg-44k dataset.

References

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

[2] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1300–1309, 2022. 4, 8, 11

[3] Anthony Cioppa, Adrien Deliège, Maxime Istasse, Christophe De Vleeschouwer, and Marc Van Droogenbroeck. Arthus: Adaptive real-time human segmentation in sports through online distillation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1, 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 4, 6

[5] Matteo Dunnhofer and Christian Micheloni. Visual tracking in camera-switching outdoor sport videos: Benchmark and baselines for skiing. *Computer Vision and Image Understanding*, 243:103978, 2024. 7, 11

[6] Matteo Dunnhofer, Luca Sordi, Niki Martinel, and Christian Micheloni. Tracking skiers from the top to the bottom. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8511–8521, 2024. 7, 11

[7] Pedro F Felzenszwalb and Daniel P Huttenlocher. Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428, 2012. 11

[8] Fang Gao, Wenjie Wu, Yan Jin, Lei Shi, and Shengheng Ma. A sparse attention pipeline for deepsportradar basketball player instance segmentation challenge. In *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports*, pages 131–135, 2023. 2

[9] Seyed Abolfazl Ghasemzadeh, Gabriel Van Zandycke, Maxime Istasse, Niels Sayez, Amirafshar Moshtaghpour, and Christophe De Vleeschouwer. Deepsportlab: a unified framework for ball detection, player instance segmentation and pose estimation in team sports scenes. *arXiv preprint arXiv:2112.00627*, 2021. 2

[10] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3129–3136, 2010. 3

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[12] Horst Karl Hahn and Heinz-Otto Peitgen. Iwt-interactive watershed transform: A hierarchical method for efficient interactive and automated segmentation of multidimensional grayscale images. In *Medical Imaging 2003: Image Processing*, volume 5032, pages 643–653. SPIE, 2003. 3

[13] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 international conference on computer vision*, pages 991–998. IEEE, 2011. 6

[14] Dongya Huang and Li Zhang. Parallel dense merging network with dilated convolutions for semantic segmentation of sports movement scene. *KSII Transactions on Internet and Information Systems (TIIS)*, 16(11):3493–3506, 2022. 2

[15] You Huang, Hao Yang, Ke Sun, Shengchuan Zhang, Liujuan Cao, Guannan Jiang, and Rongrong Ji. Interformer: Real-time interactive image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22301–22311, 2023. 3, 4, 8, 11

[16] Maxime Istasse, Vladimir Somers, Pratheeban Elancheliyan, Jaydeep De, and Davide Zambrano. Deepsportradar-v2: A multi-sport computer vision dataset for sport understandings. In *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports*, MMSports '23, page 23–29, New York, NY, USA, 2023. Association for Computing Machinery. 1, 2

[17] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5306, 2019. 3

[18] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 6, 7, 8

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 6

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv e-prints*, pages arXiv–2304, 2023. 2, 3, 4, 5, 6, 7, 8, 11

[21] Chaewon Lee, Seon-Ho Lee, and Chang-Su Kim. Mfp: Making full use of probability maps for interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4051–4059, June 2024. 11

[22] Gaston Lenczner, Adrien Chan-Hon-Tong, Bertrand Le Saux, Nicola Luminari, and Guy Le Besnerais. Dial: Deep interactive and active learning for semantic segmentation in remote sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:3376–3389, 2022. 3

[23] G. Lenczner, A. Chan-Hon-Tong, N. Luminari, B. Le Saux, and G. Le Besnerais. Interactive learning for semantic segmentation in earth observation. In *ECML-PKDD MACLEAN Workshop*, 2020. 3

[24] Jianshu Li, Jian Zhao, Yunchao Wei, Congyan Lang, Yidong Li, Terence Sim, Shuicheng Yan, and Jiashi Feng. Multiple-

human parsing in the wild. *arXiv: Computer Vision and Pattern Recognition*, 2017. 1

[25] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 5

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1, 6

[27] Qin Liu, Jaemin Cho, Mohit Bansal, and Marc Niethammer. Rethinking interactive image segmentation with low latency, high quality, and diverse prompts. *arXiv preprint arXiv:2404.00741*, 2024. 3, 5, 8

[28] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22290–22300, October 2023. 3, 4, 5, 6, 7, 8, 11

[29] Qin Liu, Meng Zheng, Benjamin Planche, Srikrishna Karanam, Terrence Chen, Marc Niethammer, and Ziyan Wu. Pseudoclick: Interactive image segmentation with click imitation. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 728–745, Cham, 2022. Springer Nature Switzerland. 11

[30] Yijia Liu, Jianhua Liu, Heng Pu, Yuan Liu, Shiran Song, et al. Instance segmentation of outdoor sports ground from high spatial resolution remote sensing imagery using the improved mask r-cnn. *International Journal of Geosciences*, 10(10):884, 2019. 2

[31] Katja Ludwig, Daniel Kienzle, Julian Lorenz, and Rainer Lienhart. Detecting arbitrary keypoints on limbs and skis with sparse partly correct segmentation masks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 461–470, 2023. 1, 2

[32] Katja Ludwig, Julian Lorenz, Robin Schön, and Rainer Lienhart. All keypoints you need: Detecting arbitrary keypoints on the body of triple, high, and long jump athletes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5178–5186, 2023. 1, 2

[33] Katja Ludwig, Sebastian Scherer, Moritz Einfalt, and Rainer Lienhart. Self-supervised learning for human pose estimation in sports. In *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2021. 1

[34] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 6

[35] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou,

Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 4, 6

[36] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 6

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 5

[38] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988, 2017. 6

[39] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, page 309–314, New York, NY, USA, 2004. Association for Computing Machinery. 3, 6

[40] Takehiro Sawahata, Alessandro Moro, Sarthak Pathak, and Kazunori Umeda. Instance segmentation-based markerless tracking of fencing sword tips. In *2024 IEEE/SICE International Symposium on System Integration (SII)*, pages 472–477. IEEE, 2024. 1, 2

[41] Robin Schön, Julian Lorenz, Katja Ludwig, and Rainer Lienhart. Adapting the segment anything model during usage in novel situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3616–3626, 2024. 3

[42] Robin Schön, Daniel Kienzle, and Rainer Lienhart. Wseg: Introducing a dataset for the segmentation of winter sports equipment with a baseline for interactive segmentation, 2024. 1, 2, 3, 6, 7, 8

[43] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8623–8632, 2020. 3

[44] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3141–3145. IEEE, 2022. 3, 4, 5, 6, 8, 11

[45] Shuhei Tarashima. Sports field recognition using deep multi-task learning. *Journal of Information Processing*, 29:328–335, 2021. 2

[46] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021. 5

[47] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong.

Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 8

A. Simulating User Clicks

In this section we discuss how we simulate the clicks. More precisely, we want to answer the question: Given a predicted mask $\mathbf{m}_\tau$ and a ground truth mask $\mathbf{m}_{GT}$, where do we place the next click in order to help our network with improving the mask? We follow common practice and use the method described in [44].

1. For each click we simulate, we compare $\mathbf{m}_\tau$ with $\mathbf{m}_{GT}$ to obtain the mask of false positives $\mathbf{m}_{FP}$ and the mask of false negatives $\mathbf{m}_{FN}$.

2. Afterwards, we compute the euclidean distance transforms (see [7]) of both masks, $\mathcal{D}_{FP}$ and $\mathcal{D}_{FN}$.

3. We will then look for the maxima of $\mathcal{D}_{FP}$ and $\mathcal{D}_{FN}$. The coordinates of the higher maximum will be the location of the simulated click.

4. Depending on whether this maximum is found in either $\mathcal{D}_{FP}$ or $\mathcal{D}_{FN}$, we will label it as a background (-) or foreground (+) click, respectively.

It should be noted that this metric allows for improving the system at the cost of practical usability. If we were to simulate the clicks during training in the exact same way as we do during testing (taking the maximum of the two distance transforms), we would prepare our model to optimally perform under the metric. This can for example be seen in [15, 21]. As [44] however mentions, this inhibits the practical usability of the model, since an actual human would choose other non-optimal click positions. We would see a kind of overfitting to the metric. To make sure the training of our model adheres to practical requirements, we follow common practice [2,20,28,29,44] and use additional random clicks during each training step.

B. Changing the Number of Encoder Blocks after Adding the Prompts

The prompt features and the image features in our architecture are fused by multiple transformer encoder blocks. In our standard model we chose four as the number of blocks. In Tab. 6 we compare the performance of the model when altering the number of blocks (the column *Depth*). We cannot observe a clear trend, as a continuous increase of the number of blocks does not necessarily cause an improvement. We even see that reducing the number of blocks to three gives a slightly better performance for a NoC@85 of 6.94 to 6.31, although the best performance depends on the metric, with 6.31 for the NoC@85 and 9.023 for the NoC@90.

Depth	WSESeg Average	
	NoC@85	NoC@90
2	6.962	9.587
3	**6.311**	9.091
4	6.944	9.163
5	6.689	9.310
6	6.524	**9.023**

Table 6. A comparison of the change in performance for different numbers of ViT blocks. The *depth* does not refer to the backbone, but the additional blocks after mixing the image and prompt features. The NoC is the average over all classes.

C. Qualitative Examples from SHSeg

In Figure 4, we can see qualitative examples from our newly proposed SHSeg (Skiing Human Segmentation) dataset. Our dataset provides 534 masks for skiers on 496 images. The images have been randomly sampled from the SkiTB dataset [5,6]. A link to the data can be found in our main paper (publication of data upon acceptance).
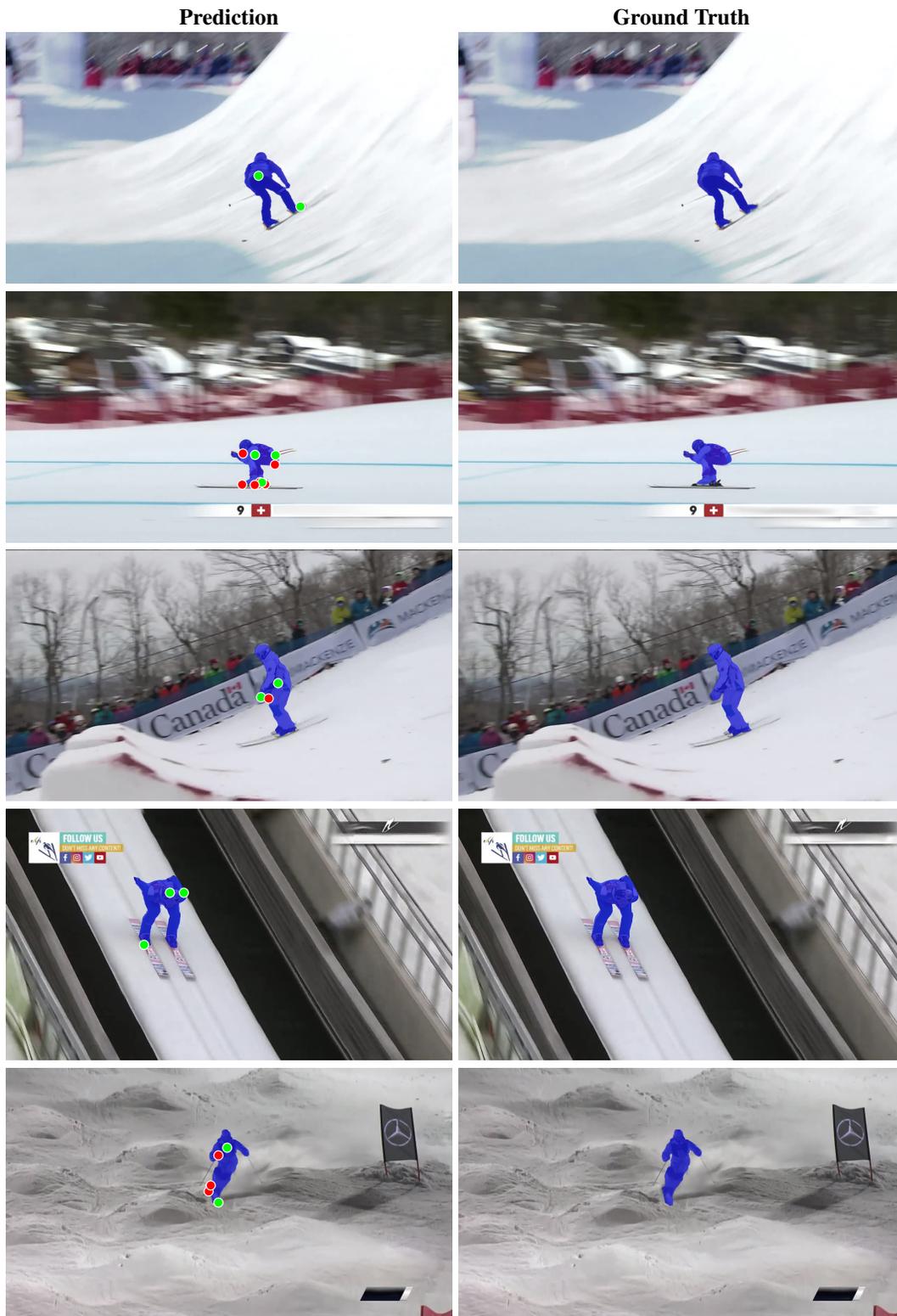
Figure 4. Examples for the masks occurring during the interaction. The *left column* displays the predicted mask along with the clicks. Foreground clicks are *green*, background clicks are *red* and the masks are *blue*. The *right column* displays the corresponding ground truth.