# Interactive Segmentation for Diverse Gesture Types Without Context

Josh Myers-Dean[1], Yifei Fan[2], Brian Price[2], Wilson Chan[2], and Danna Gurari[1,3]
[1]University of Colorado Boulder [2]Adobe Research [3]University of Texas at Austin

## Abstract

*Interactive segmentation entails a human marking an image to guide how a model either creates or edits a segmentation. Our work addresses limitations of existing methods: they either only support one gesture type for marking an image (e.g., either clicks or scribbles) or require knowledge of the gesture type being employed, and require specifying whether marked regions should be included versus excluded in the final segmentation. We instead propose a simplified interactive segmentation task where a user only must mark an image, where the input can be of any gesture type without specifying the gesture type. We support this new task by introducing the first interactive segmentation dataset with multiple gesture types as well as a new evaluation metric capable of holistically evaluating interactive segmentation algorithms. We then analyze numerous interactive segmentation algorithms, including ones adapted for our novel task. While we observe promising performance overall, we also highlight areas for future improvement. To facilitate further extensions of this work, we publicly share our new dataset at https://github.com/joshmyersdean/dig.*

## 1. Introduction

A common goal is to locate regions, such as objects or object parts, in images. We refer to this task as *region segmentation*. A challenge is that fully-automated solutions often are error-prone while exclusive reliance on human annotations is costly and time-consuming. As a middle ground, *interactive segmentation* methods empower humans to supply minimal input towards collecting *consistently high-quality region segmentations*. Two popular interactive segmentation settings are to generate a segmentation (1) when the only input is a human marking on an image (i.e., **segmentation creation**) and (2) when the input is a human marking and a previous segmentation (i.e., **segmentation refinement**).

While existing interactive region segmentation methods are widely-used and beneficial to society, they have at least one of the following two important limitations. The first centers on how users interact with the methods. For most methods, a single *gesture type* is supported as human input,

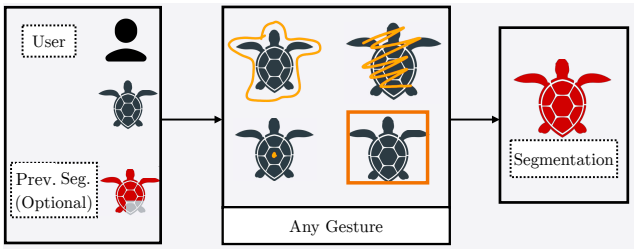



Figure 1. Overview of proposed interactive segmentation task. A user can mark an image with any gesture type which a method then uses to either create a segmentation from scratch or to refine a previous (imperfect) segmentation (if available). This is done without any further guidance, including without specifying whether that marked region is content to include versus exclude.

such as either only clicks [10, 18, 42, 54, 62], scribbles [3, 15, 30, 33, 63], lassos [5, 38, 47, 52, 61], extreme points [44, 50], or rectangles [8, 51, 61, 64, 65]. Yet, we will show findings from a user study in Section 3 that users prefer different gestures in different scenarios. The second limitation is that all methods, except a few that only can accept a single gesture type (e.g., [61]), require users to provide *context* of whether the content they annotate should be included in or excluded from the final segmentation. This effort imposes an extra burden on humans that we hypothesize is unnecessary.

We propose a new interactive segmentation task where an algorithm *only takes as input a gesture of any type*. An overview is shown in Figure 1. This novel task relieves users from the burden of having to either learn how to use different tools for different gesture types or micro-manage how an algorithm interprets user input, such as signifying gesture type [25, 37] or context of whether to include versus exclude the marked region [10, 25, 37, 44, 54, 62]. We call this novel task *gesture-agnostic, context-free interactive segmentation*.

We make several contributions in support of the novel task. First, we present the first interactive segmentation dataset for explicitly training and evaluating models to support *multiple gesture types*. We call it the **D**iverse **I**nteractive **G**esture (DIG) dataset and release it publicly to encourage community-wide progress. Second, we propose an evaluation metric to holistically evaluate algorithms for our new task. Finally, we benchmark the benefits of modern interac-

tive segmentation algorithms for our proposed task, including after adapting their algorithmic frameworks for our task. While we observe promising performance overall, we also highlight areas for future improvement.

Success on this new task can yield numerous benefits to society. It would accelerate image editing, empowering users to simply use the gesture most natural to them when creating segmentations without becoming an expert on a vast array of 'features' that support different gesture types and specifications for context. This could directly benefit lay users of image editing applications (e.g., Photoshop [2]) as well as specialized practitioners who depend on segmentations for downstream analysis (e.g., doctors performing diagnoses in the medical community [29, 56, 57]). Such methods would also support more efficient curation of labeled training data to develop region segmentation models [1, 4, 7, 25]. Finally, our work sets a precedent that could be generalized to segmentation tasks beyond region segmentation. For example, future studies could explore *gesture-agnostic, context-free interactive segmentation* for tasks such as semantic segmentation [20, 58, 60] and panoptic segmentation [24, 40, 46].

## 2. Related Work

**Interactions in Interactive Segmentation Methods.** Interactive segmentation methods typically accept up to two types of information from users. First, annotations of a region in an image are given to a model through gestures such as scribbles [3, 15, 30, 33, 55, 63], lassos [5, 52, 61], extreme points [44, 50], rectangles [8, 25, 51, 61, 64, 65], and (most commonly) clicks [9, 25, 26, 31, 36, 43]. For the subset of algorithms that can support multiple gesture types, they require additional input. For example, Segment Anything (SAM) [25] requires contextual information during click interactions and specification of the gesture type to decide which prompt encoder (i.e., click, rectangle, or text) to utilize. Similarly, Multi-Mode Interactive Segmentation (MMIS) [37] requires context as well as a bounding gesture (e.g., lasso, rectangle) for the initial segmentation before supporting non-bounding gestures (e.g., clicks). The second type of input common for interactive segmentation methods [7, 10, 14, 25, 42, 54] is context as to whether annotated content should be included or excluded in the final segmentation. Only a few methods do not require this input. Some assume marked content should always be included (e.g., Deep GrabCut [61]). Alternatively, language can be used in place of gestures. While useful for many computer vision tasks [17, 39] and potentially valuable in some interactive segmentation situations, relying only on language has important limitations. Not only do many vision-language models have poor multilingual support [6, 11, 23, 41]), but also often it can be difficult to articulate what one wishes to edit (e.g., a small correction to an existing segmentation). Extending prior work, we propose a simplified task of *gesture-agnostic, context-free interactive segmentation*. It is less cumbersome than the status quo as it reduces human effort to only a single input: marking an image with any gesture.

**Interactive Segmentation Datasets.** Most interactive segmentation methods are trained and/or evaluated on repurposed datasets originally designed for other tasks, such as semantic/instance segmentation [4, 13, 16, 19, 27, 34], salient object segmentation [45, 51], entity segmentation [25], or video segmentation [10, 48]. For example, COCO+LVIS [54] combines two popular object segmentation datasets (i.e., COCO [34] and LVIS [16]) and is repurposed for interactive segmentation by removing semantic/instance information. DAVIS-585 [10] samples frames from the video segmentation dataset DAVIS [49], resulting in 300 images and 585 object segmentation annotations. The one exception is the SA-1B dataset [25], which was built using a large-scale human-machine collaboration to collect entity labels, resulting in 1.1 million images with 1 billion masks. Unlike existing datasets, we introduce the first dataset that comes with predefined gestures of different types and so enables developing one-size-fits-all algorithms to address the *gesture-agnostic, context-free interactive segmentation* problem.

**Interactive Segmentation Evaluation Metrics.** The research community has employed a variety of evaluation metrics to assess the performance of interactive segmentation methods. For example, some works rely on traditional measures for segmentation evaluation, including intersection-over-union (IoU) [32, 33, 44, 62] and mean average precision (mAP) [61]. Alternatively, most click-based interactive segmentation methods [10, 35, 36, 53, 54] use the number of clicks (NoC), which captures how many clicks a model requires on average to achieve a target IoU. A limitation of existing metrics is that, for methods that refine a previous segmentation, they fail to capture the extent to which resulting segmentations are better (or worse) than the previous segmentation. We propose the first metric that can capture the relative change of a segmentation from a previous state, including an empty mask for segmentation creation.

## 3. User Study on Gesture Type Preferences

We conducted a user study to establish what types of gestures people naturally gravitate towards when using interactive segmentation methods. To our knowledge, no prior work has published such a study.

**Study design.** We designed 42 segmentation tasks reflecting a range of scenarios including segmenting salient objects, semantic categories, and multiple objects of different shapes (e.g., thin, occluded). Each segmentation task consisted of an image and short instruction at the top asking the participant to select something in the image. Participants were told, "Use whatever gesture(s) that feel natural to you to make

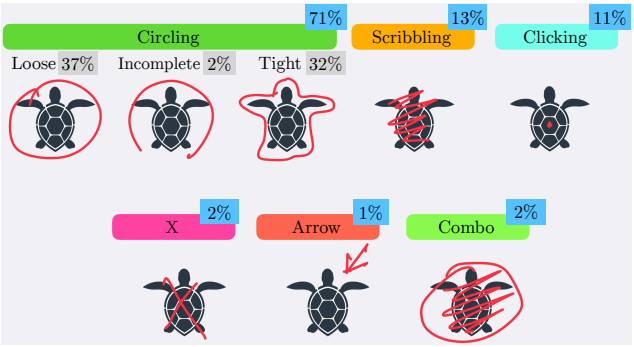



Figure 2. Breakdown of gesture frequency from our user study (N=1795) and examples of gestures. The observed diversity motivates designing algorithms to support multiple gesture types.

the selection asked: clicking, circling, drawing an outline, drawing a rectangle, drawing a polygon, scribbling, drawing a line, etc." To continue on to the next task, participants were instructed, "After you are done making the gesture(s), you can move on to the next task. You will not receive any feedback from the app." Study participants were given the option to use either their fingers or a stylus when drawing a gesture. The study was conducted on an iPad with iPadOS 14 or above using an unbranded application in landscape mode with a scrollable and zoomable canvas.

**Study participants.** We recruited participants from UserTesting[1] who identified as having either novice or intermediate photo editing skills. In total, we had 43 participants. Our participants reflected a range of demographics, including participants between the ages of 18 and 55 and a balanced gender ratio. Each participant was shown the 42 segmentation tasks in a randomized order.

**Results.** Overall, we collected 1795 gesture annotations (i.e., 42 segmentation tasks $\times$ 43 participants with some participants not completing all tasks). We tallied the type of gesture used based on the following gesture type categories: circling (i.e, lassos of varying granularity), scribbling, clicking, X's, arrows, and any combination of gestures. Gesture labels were assigned by visual inspection. Figure 2 shows the results of the user study.

We found that a diversity of gestures were used. Lassos were most common and varied between tight and loose. From further inspection, we found that different gestures were preferred in different contexts. For example, when selecting power lines in one image (i.e., a thin object based on visual inspection), participants used scribbles 54% of the time and lassos 40% of the time. In another task where participants were meant to select a chain link fence (i.e., another thin object), participants used scribbles 56% of the time and lassos 26% of the time. For another task, participants were instructed to select repeated sprinkles on a donut; participants used clicks 51% of the time and lassos 28%. In a task where participants needed to select a simple background (i.e., a wall with a woman in the foreground), participants used lassos 42% of the time, clicking 26% of the time, scribbles 21% of the time, and X's or arrows 7% of the time.

Overall, our findings underscore the value of one-size-fits-all interactive segmentation models that support a variety of gesture types to enable a more seamless and user-friendly experience for diverse users. Of note, great variation in gestures may also be observed on axes we did not test. For example, we only tested with an iPad. Gesture preference may vary if the user is using a mouse on a desktop, a trackpad on a laptop, or their finger on a small smartphone screen where their finger will occlude the target object. Variation may also exist across age groups or among people with motor or visual impairments [22]. This observation further motivates the need for our new approach to interactive segmentation, and we now discuss our work to establish this research direction.

## 4. DIG Dataset

We now present the **D**iverse **I**nteractive **G**esture (DIG) dataset, which supports two popular settings: generating a segmentation from only a marking on an image (i.e., **segmentation creation**) and generating a segmentation from an image, previous segmentation, and marking (i.e., **segmentation refinement**). To our knowledge, DIG is the first interactive segmentation dataset with multiple gesture types.

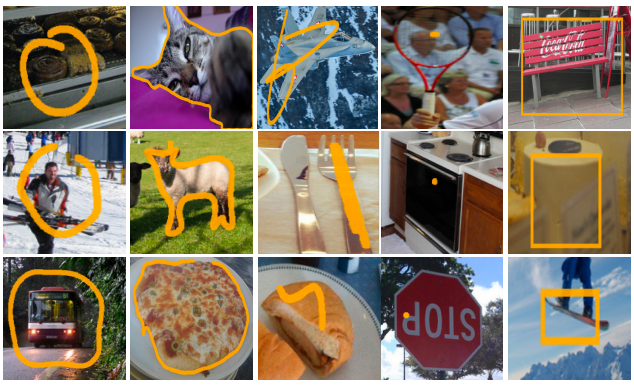
### 4.1. Dataset Creation

**Image Source.** We leverage 103,902 images from COCO+LVIS [54], a popular source for interactive segmentation [10, 18, 54]. Those images are advantageous because they each contain multiple objects, which in turn means that algorithms cannot simply learn to locate salient objects. An additional strength is that it includes a long tail of object types, a key motivation for creating LVIS [16].

**Ground Truth Region Segmentations.** We use the same source for the ground truth of region segmentations as our dataset source (i.e., COCO+LVIS [54]). Specifically, the ground truths are derived from instance-level segmentations. Consequently, we disregard semantic information since we are only concerned with locating a region within an image. To generate ground truth for selecting *part* of a region, we exploit that some regions in COCO+LVIS are broken into more than one region by occlusions. In such cases, we select one sub-region (i.e., connected component) as ground truth.

**Dataset Filtering.** Following prior work on interactive segmentation datasets [10], we remove objects with an area of fewer than 300 pixels. This acknowledges that such objects occupy a tiny portion of an image (e.g., at most 0.001% of

[1] https://www.usertesting.com/




(a) (b) (c) (d) (e)

Figure 3. Examples of gesture types in DIG: (a) loose lassos, (b) tight lassos, (c) scribbles, (d) clicks, and (e) rectangles. Images are cropped to the target region for visualization.

the area assuming a 512x512 resolution) and also that small areas are challenging for modern neural networks to identify due to information loss from operations such as downsampling and max pooling. Our final dataset consists of 103,902 images with 886,612 regions and 194,855 parts.

**Previous Segmentation Generation.** To enable fair, reproducible algorithm benchmarking for the interactive setting of refining a previous segmentation, we supply initial segmentations. We employ the superpixel-based approach introduced in [10]. For this interactive segmentation setting, we also consider segmenting *multiple disconnected regions* (e.g., a banana and an orange sitting on a table). We introduce this additional setting in order to facilitate teaching algorithms through training data to consider the relationship between the gesture and the underlying image content rather than only the gesture and the previous segmentation. Details can be found in the Supplemental Materials.

**Gesture Annotations.** We generate a static set of gesture annotations to enable consistent, fair algorithm comparison.[2] Motivated by our user study findings (Section 3), we focus on the following gestures most commonly observed in practice: lassos, scribbles, and clicks. We also augment rectangles because they are a popular gesture in industry applications, such as the Rectangle Marquee tool in Photoshop [2], and can lead to faster whole-object coverage than lassos [21]. We generate annotations corresponding to multiple gesture types for each region in our dataset, as exemplified in Figure 3. A summary of how we construct each gesture is described below, and technical details are in the Supplementary Materials.

For **lasso generation**, we create both loose and tight lasso gestures to accommodate variable user behaviors. For example, in some cases, a user may lack a steady hand, leading to imperfect or noisy lasso boundaries. We construct tight lassos by interpolating points sampled from a region boundary and "jitter" these points to simulate user noise.

For **scribble generation**, we capture scribbles ranging from highly curved, squiggles to smooth curves. We randomly sample points from the target region or previous segmentation and then interpolate between them to create a B-spline curve followed by mechanisms to simplify curves and have scribbles pass outside the region's boundaries.

For **click generation**, we account for multiple click locations, spanning from each region's center to near its boundary. Like prior work [62], we randomly select a foreground pixel (when creating a segmentation) or a pixel from a previous segmentation (when refining a segmentation).

For **rectangle generation**, we use the approach presented by [61]. Given a tight bounding box, we modify the corners to perturb how closely the box encloses the region of interest.

### 4.2. Dataset Analysis

We now characterize DIG and how it compares to seven existing interactive segmentation datasets: DAVIS-585 [10], COCO+LVIS [54], GrabCut [28, 51], Berkeley [45], PASCAL+SBD [13, 19], SA-1B [25], and OpenImages [4, 27]. Those seven represent datasets that are long-standing interactive segmentation benchmarks for the research community (Berkeley, GrabCut), large-scale (COCO+LVIS, OpenImages, SA-1B), and popular for training interactive segmentation methods (COCO+LVIS, PASCAL+SBD, SA-1B). For each dataset, we report the types of gestures included, whether previous segmentations for refinement are provided, number of images, number of unique regions, and total number of samples (i.e., number of unique gesture-region combinations). Results are shown in Table 1.

Our dataset is the only one providing pre-computed annotations of *multiple gesture types*. The only other dataset supplying pre-computed interactive gestures is GrabCut [28], and only for 'tight' rectangles (bounding boxes). Consequently, our dataset is the first to support efficient training and evaluation of gesture-agnostic interactive segmentation algorithms by providing a standardized set of annotations that enables uniform comparison of algorithms.

Another distinction of our dataset is that it provides previous segmentations to support algorithm benchmarking for the segmentation refinement setting. Only one other dataset supports this setting, DAVIS-585, but only with a testing split, thereby lacking splits for algorithm development.

A further distinction is that DIG has at least four times as many samples as all datasets, except SA-1B [25]. This arises primarily because our dataset includes multiple gesture types per each region in every image. We expect this to benefit deep learning methods (i.e., the de facto tool for interactive segmentation), which need large amounts of training data. While SA-1B contains more samples than DIG,

[2]While one may consider generating gestures on-the-fly, this is not only inefficient but also impractical, as discussed in the Supplementary Materials.

| Dataset | DIG | DAVIS-585 [10] | COCO+LVIS [54] | GrabCut [28, 51] | Berkeley [45] | PASCAL+SBD [13, 19] | OpenImages [4, 27] | SA-1B [25] |
|---|---|---|---|---|---|---|---|---|
| **Gesture Types** | L,C,S,R | - | - | R | - | - | - | - |
| **Prior Seg** | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| **# Images** | 104K | 300 | 104K | 50 | 300 | 11.5K | 1M | 11M |
| **# Regions** | 1M | 585 | 1.5M | 50 | 300 | 31K | 2.8M | 1.1B |
| **# Samples** | 13.5M | 585 | 1.5M | 50 | 300 | 31K | 2.8M | 1.1B |

Table 1. Comparison of DIG to six interactive segmentation datasets. DIG is the only dataset to support multiple gesture types. We report the approximate number of samples rounded to the nearest order of magnitude. (L=Lasso, R=Rectangle, C=Click, S=Scribble)

it does not provide pre-computed previous segmentations or interaction annotations to enable consistent, and so fair, algorithm comparison.

## 5. Evaluation Metric: RICE

We now introduce a new evaluation metric for interactive segmentation to address that existing metrics (e.g., NoC, IoU) do not capture the amount that a method worsens/improves the results when refining an initial segmentation. For example, they could not capture that a result *worsens* when a segmentation has a lower IoU with the ground truth than a previous segmentation. Similarly, existing metrics cannot distinguish a relatively smaller change from the previous segmentation compared to a large one such as when a model produces a prediction that has an IoU of 95% when refining a previous segmentation with a high initial IoU (e.g., 90%) versus a low initial IoU (e.g., 20%). We propose the first metric to holistically evaluate interactive segmentation algorithms regardless of gesture type and segmentation setting.

We call our metric the **R**elative **I**oU **C**orrective **E**valuation metric, or RICE. RICE takes into consideration how well a predicted segmentation improves/damages a previous segmentation with respect to a region's ground truth and simplifies to IoU when no previous segmentation is present. Formally, we define RICE as:

$$RICE(\alpha, \beta) = \begin{cases} \frac{\alpha-\beta}{1-\beta}, & \text{if } \alpha \geq \beta \\ \frac{\alpha}{\beta} - 1, & \text{else} \end{cases}, \qquad (1)$$

where $\alpha$ is the IoU between a region's ground truth and the output of an interactive segmentation model after thresholding. $\beta$ is the IoU between a previous segmentation and a region's ground truth. Values can be positive or negative, where a negative value indicates there was a previous segmentation and, in the attempt to correct a mistake, the model prediction *decreased* the overall IoU with the region ground truth rather than increasing it as desired. Analysis highlighting RICE's derivation and intuition as well as its benefits over IoU are provided in the Supplementary Materials.

## 6. Algorithm Benchmarking

We next benchmark modern interactive segmentation methods as is as well as adapted for our novel task. We perform all experiments on an NVIDIA A100 GPU.

**Dataset Splits.** We leverage the splits used in the COCO+LVIS dataset [54] to divide DIG into training, validation, and test splits, since DIG is built upon that dataset. We assign all training images from COCO+LVIS to our training dataset and then split the images in the validation set for COCO+LVIS into validation and testing splits using a random 70%/30% split. Our final dataset consists of a: *training set* with 99,161 images, 857,669 regions, 186,740 region parts, and 12,839,936 samples; *validation set* with 3,318 images, 32,938 regions, 5698 region parts, and 413,784 samples; and *test set* with 1,423 images, 11,703 regions, 2417 region parts, and 246,080 samples. Of the 246,080 samples in the test set, 69,852 samples belong to the scenario of *multi-region* segmentation, as described in Section 4.1.

**Baseline Models.** Despite that algorithms do not exist to support our novel task, as all interactive segmentation methods require more input than our task permits, we still aim to gauge how well existing state-of-the-art interactive segmentation methods could work if modified for our task. We evaluate these top-performing algorithms that only support one gesture type since their code is publicly-available:

- *Deep GrabCut [61]:* top-performing model that takes as human input rectangles. It supports segmentation creation.
- *IOG: [65]:* top-performing model that takes as human input a tight bounding box with a central click. It supports segmentation creation.[3]
- *RITM [54]:* second best performing non-foundation model on four datasets for taking in human input click gestures. This method supports segmentation creation and refinement. We use the HRNet [59]-18s variant.
- *FocalClick [10]:* top performing non-foundation model on four datasets for taking as human input click gestures. This method supports segmentation creation and refinement. We use the HRNet [59]-18s variant.[4]

To adapt these models for gesture-agnostic, *context-free* interactive segmentation, as they do not explicitly support

[3] IOG [65] refines its own predictions, not arbitrary previous ones, so it's unsuitable for comparison in our setting.

[4] We omit the refinement module as we observed worse performance with it; details are provided in the Supplementary Materials.

other gesture types, we convert all gesture types into the type supported by each model. For evaluation, since providing context to the models is incompatible with our proposed task, we instead assess each model using three approaches for supplying context:

- *positive:* for each click, the method only receives content as 'positive' (i.e., content should be included).
- *negative:* for each click, the method only receives content as 'negative' (i.e., content should not be included).
- *random:* for each click, the method receives context decided by a draw from a discrete uniform distribution between positive and negative context.

We also evaluate the state-of-the-art model that supports multiple gesture types, SAM [25].[5] Its official implementation[6] supports clicks (context *required*) and rectangles (context *not required*). For other gesture types (i.e., scribbles and lassos), we either encode the points that compose the gesture as clicks (i.e., *SAM-C*) or as rectangles (i.e., *SAM-R*). We utilize the ViT-H [12] variant of SAM and select the output mask (out of three) with the highest IoU with ground truth.

**Proposed Models.** We also introduce models that, by design, support multiple gesture types while only accepting as input gesture annotations.[7] We adapt the HRNet [59]-18s variant of FocalClick [10] since it is both appropriate for use on a variety of devices [10, 54] because HRNet-18s is lightweight and also because it is the state-of-the-art for click-based segmentation outside of the parameter-heavy foundation models (e.g., SAM). We introduce two variants, which we call *HRNet-base*, *HRNet-dataAug*. Of note, these models will exemplify the advantage possible when training on our proposed DIG dataset.

*HRNet-base* is the original algorithm with two modifications. First, while FocalClick [10] takes as input a concatenation of a channel for positive interactions (context), a channel for negative interactions (context), and a previous segmentation (that may be blank), we instead have it take as input a concatenation of the gesture, a Euclidean distance map, and a previous segmentation. We exclude the positive and negative interactions because our problem does not permit the context of an interaction. We introduce the Euclidean distance map because it has be shown to improve results when context is not present [61].

*HRNet-dataAug* is *HRNet-base* with data augmentation to encourage algorithms to learn the relationship between a gesture and a region rather than a region alone. To augment data, given an object with no previous segmentation, we set another image region as a previous segmentation and include it in the final ground truth with probability $p$, with $p = 0.2$.

**Evaluation Metric.** We use our proposed RICE metric to evaluate both globally and locally.[8] The *global* metric is defined to measure how well an algorithm's prediction matches the region's ground truth. For our *local* metric, we only consider how well the algorithm fixes a single connected mistake targeted by the gesture.[9] This is based in part on the observation that segmentation mistakes can occur on different parts of a region, causing multiple spatially disconnected groups of pixels that need to be corrected, and those disconnected mistakes will generally be corrected by users one at a time. Similarly, for segmentation creation, the local metric establishes how well an algorithm selects a *region part*, such as the head of a dog compared to it's whole body.

**Experimental Design.** Given an interaction (i.e., click, scribble, lasso, rectangle), we examine the RICE score for that gesture in both segmentation settings (i.e., with a previous segmentation and without). Due to time and computation constraints, we do not consider combinations of multiple gestures for evaluation as there are $\binom{5}{n}$ possible combinations for every region in the test set, where $n$ is the number of gestures chosen. An additional practical reason is that in an interactive session, the user will typically receive feedback from one marking before making the next marking.

**Overall Results.** Results are shown in Table 2. We report a breakdown of these results, all results for poor performing models (i.e., Deep GrabCut [61] and IOG [65]), and multi-region segmentation results (which follow trends for our top-performing segmentation refinement method) in the Supplemental Materials.

As shown, models that support multiple gesture types outperform methods that support a single gesture type. Within the top-performing models that support multiple gesture types simultaneously, we observe mixed outcomes. For the segmentation creation setting, *SAM-R - positive* (i.e., interactions mapped to rectangles with all context assumed to be positive) performs the best. However, this improvement comes at the cost that SAM-R requires the gesture type as input and is parameter-heavy. In contrast, our simplified model with data augmentation (*HRNet-dataAug*) achieves nearly comparable performance (i.e., 1.45 percentage points behind

[5] MMIS [37] is not evaluated since the code was not publicly available at the time of submitting this paper.

[6] https://github.com/facebookresearch/segment-anything

[7] For completeness, we also discuss two variants in the Supplementary Materials that highlight the performance of our adapted framework when permitting context. Overall, neither variant leads to a performance boost.

[8] To be backwards compatible in evaluation, we conduct an additional assessment that is discussed in the Supplementary Materials due to space constraints. We evaluate algorithms for three IoU thresholds on DAVIS585 [10] using the prior standard evaluation metric, number of clicks, adapted to number of gestures to accommodate multiple gesture types. Our findings show that context-augmented algorithms underperform and need more interactions compared to our proposed models.

[9] As outlined in the Supplementary Materials, our evaluation method is more realistic than previous approaches (e.g., [33, 62, 65]). Unlike click-based methods that center clicks around the largest error, our interactions are varied in position, as shown in Figure 3.

| | Method | Average | | Click | | Scribble | | Loose Lasso | | Tight Lasso | | Rectangle | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $RICE_{local}$ | $RICE_{global}$ | $RICE_{local}$ | $RICE_{global}$ | $RICE_{local}$ | $RICE_{global}$ | $RICE_{local}$ | $RICE_{global}$ | $RICE_{local}$ | $RICE_{global}$ | $RICE_{local}$ | $RICE_{global}$ |
| Creation | *RITM [54] - positive* | 29.83 | 28.90 | 54.03 | 52.78 | 45.01 | 43.15 | 1.28 | 1.23 | 32.96 | 31.78 | 15.88 | 15.55 |
| | *RITM [54] - negative* | 10.68 | 10.45 | 0.00 | 0.00 | 1.17 | 1.18 | 11.23 | 11.23 | 14.34 | 13.92 | 26.64 | 25.94 |
| | *RITM [54] - random* | 20.33 | 19.74 | 27.15 | 26.47 | 23.10 | 22.13 | 6.32 | 6.30 | 23.56 | 22.78 | 21.49 | 21.00 |
| | *FocalClick [10] - positive* | 28.92 | 28.03 | 54.95 | 53.47 | 44.78 | 43.12 | 1.29 | 1.26 | 29.49 | 28.44 | 14.11 | 13.86 |
| | *FocalClick [10] - negative* | 9.01 | 8.91 | 0.13 | 0.16 | 0.70 | 0.74 | 12.01 | 12.03 | 9.92 | 9.87 | 22.29 | 21.78 |
| | *FocalClick [10] - random* | 18.95 | 18.50 | 27.83 | 27.07 | 22.86 | 22.02 | 6.51 | 6.60 | 19.38 | 19.07 | 18.15 | 17.75 |
| | *SAM [25]-R - positive* | **67.25** | **65.44** | **77.94** | **77.26** | 63.20 | 60.51 | 41.63 | 40.97 | 74.25 | 71.83 | **79.22** | **76.63** |
| | *SAM [25]-R - negative* | 56.21 | 54.39 | 22.73 | 22.00 | 63.20 | 60.51 | 41.63 | 40.97 | 74.25 | 71.83 | **79.22** | **76.63** |
| | *SAM [25]-R - random* | 61.75 | 59.93 | 50.42 | 49.70 | 63.20 | 60.51 | 41.63 | 40.97 | 74.25 | 71.83 | **79.22** | **76.63** |
| | *SAM [25]-C - positive* | 55.54 | 54.39 | **77.94** | **77.26** | **66.83** | **64.85** | 8.49 | 8.64 | 45.20 | 44.55 | **79.22** | **76.63** |
| | *SAM [25]-C - negative* | 33.90 | 32.99 | 22.73 | 22.00 | 20.78 | 20.13 | 9.59 | 9.69 | 37.19 | 36.52 | **79.22** | **76.63** |
| | *SAM [25]-C - random* | 44.67 | 43.63 | 50.03 | 49.36 | 43.98 | 42.62 | 9.05 | 9.18 | 41.06 | 40.37 | **79.22** | **76.63** |
| | *HRNet-base* | 64.37 | 61.84 | 54.85 | 52.02 | 59.55 | 56.99 | 67.24 | 65.66 | 80.88 | 77.25 | 59.33 | 57.29 |
| | *HRNet-dataAug* | 66.62 | 63.99 | 57.81 | 54.83 | 61.25 | 58.59 | **69.15** | **67.51** | **82.12** | **78.40** | 62.79 | 60.65 |
| Refinement | *RITM [54] - positive* | -18.59 | -11.13 | -1.49 | 10.15 | -3.74 | 5.84 | -43.84 | -40.61 | -24.50 | -17.89 | -19.36 | -13.13 |
| | *RITM [54] - negative* | -16.28 | -7.52 | -2.19 | 9.49 | -5.25 | 5.64 | -24.72 | -19.04 | -23.32 | -15.13 | -25.91 | -18.56 |
| | *RITM [54] - random* | -17.47 | -9.35 | -1.84 | 9.80 | -4.56 | 5.74 | -34.13 | -29.68 | -24.06 | -16.64 | -22.75 | -15.95 |
| | *FocalClick [10] - positive* | -28.50 | -15.69 | -14.17 | 1.71 | -13.11 | 2.79 | -56.34 | -49.25 | -31.66 | -18.68 | -27.21 | -15.03 |
| | *FocalClick [10] - negative* | -24.56 | -11.68 | -7.28 | 11.85 | -14.62 | 1.16 | -36.89 | -29.41 | -30.23 | -18.70 | -33.79 | -23.29 |
| | *FocalClick [10] - random* | -26.55 | -13.76 | -10.70 | 6.78 | -13.89 | 1.83 | -46.80 | -39.60 | -30.83 | -18.58 | -30.51 | -19.23 |
| | *SAM [25]-R - positive* | 20.95 | 21.31 | 42.40 | 44.22 | 6.94 | 6.89 | 25.79 | 25.92 | 18.81 | 18.82 | 10.81 | 10.71 |
| | *SAM [25]-R - negative* | -83.78 | -82.09 | -94.86 | -93.82 | -91.93 | -91.51 | -68.59 | -64.49 | -76.46 | -74.28 | -87.07 | -86.33 |
| | *SAM [25]-R - random* | 18.31 | 18.54 | 29.22 | 30.38 | 6.94 | 6.89 | 25.79 | 25.92 | 18.81 | 18.82 | 10.81 | 10.71 |
| | *SAM [25]-C - positive* | -72.56 | -65.49 | -50.02 | -32.95 | -62.95 | -54.19 | -84.24 | -80.72 | -78.40 | -73.16 | -87.18 | -86.45 |
| | *SAM [25]-C - negative* | -90.42 | -89.20 | -94.86 | -93.82 | -92.42 | -90.86 | -88.40 | -87.11 | -89.21 | -87.78 | -87.18 | -86.45 |
| | *SAM [25]-C - random* | -81.51 | -77.35 | -72.49 | -63.34 | -77.85 | -72.69 | -86.19 | -83.74 | -83.84 | -80.51 | -87.18 | -86.45 |
| | *HRNet-base* | 36.94 | 44.84 | 34.45 | 44.18 | 36.29 | 43.54 | 35.94 | 43.71 | 40.90 | 44.79 | 37.13 | 41.22 |
| | *HRNet-dataAug* | **38.11** | **50.18** | **36.26** | **50.94** | **37.91** | **50.02** | **38.11** | **51.44** | **40.95** | **51.06** | **37.33** | **48.22** |

Table 2. Results on the test set of DIG. Above the dashed line represents segmentation creation and below represents segmentation refinement. SAM [25] performs the best during segmentation creation while *HRet-dataAug* the best during refinement.

*SAM-R - positive* with respect to $RICE_{global}$), while having 99.56% fewer parameters and no extra input requirements. For the segmentation refinement setting, *HRNet-dataAug* performs the best overall. In contrast, SAM-C [25] has the worst results, likely due to the problem that SAM discards the relevant mask when the first interaction has negative context. Overall, we contend that the advantages of the less cumbersome, more lightweight *HRNet-dataAug* outweigh those of SAM based models.

For methods that only support clicks (i.e., RITM, FocalClick), not only do they require additional input information compared to our proposed models, but they also perform worse. For example, the most effective single-gesture approach (i.e., *FocalClick - positive*) performs 2.86 percentage points worse than our baseline with respect to the $RICE_{local}$ score for clicks during segmentation creation. Similarly, it performs 1.36 percentage points worse with respect to $RICE_{global}$. We hypothesize this is partly because click-based methods are tailored for optimizing for the different metric of NoC for scenarios with multiple sequential interactions.

Qualitative results are shown in Figure 4 for segmentation creation. Results for single-gesture methods (i.e., top two rows) reinforce the quantitative findings that they struggle to segment the region of interest. As exemplified, a plausible reason for their shortcomings is that gestures such as lassos and bounding boxes have the context outside or at the boundary of the region. While we observe a similar pattern for SAM [25]-C, we find this issue is resolved when using instead rectangle encodings (i.e., SAM-R). This is likely because, unlike SAM-C, SAM-R does not rely on contextual information. We also observe that in Figure 4(b) that only the multi-gesture methods target the shirt of the baseball player rather than the entire player. We suspect this is due to the inclusion of annotations for region parts in the datasets used for training, including from our DIG dataset. A further encouraging outcome is that when the image marking targets the entirety of the desired region (i.e., using lassos), then the proposed appropriately segment the entire person.

**Analysis With Respect to Gesture Type.** There is a disparity in the performance of different gesture types across methods. Among *single-gesture methods*, clicks yield the most favorable results, while loose lassos display the least effectiveness. This observation is likely due to the training approach, where the interactions in single-gesture methods are typically centered around the region of interest. In contrast, loose lassos typically fall outside the region of interest, although they may intersect with it due to the boundary sampling and interpolation methods discussed in Section 4.1. For *multi-gesture* methods such as SAM [25], we find that clicks and rectangles yield similar results during segmentation creation with positive context, likely due to the fact that these are the gestures that are natively supported by SAM, requiring no separate encoding.

In contrast to methods that take in context, our HRNet variants show better performance with tight lassos across all evaluation metrics, while clicks tend to yield poorer results

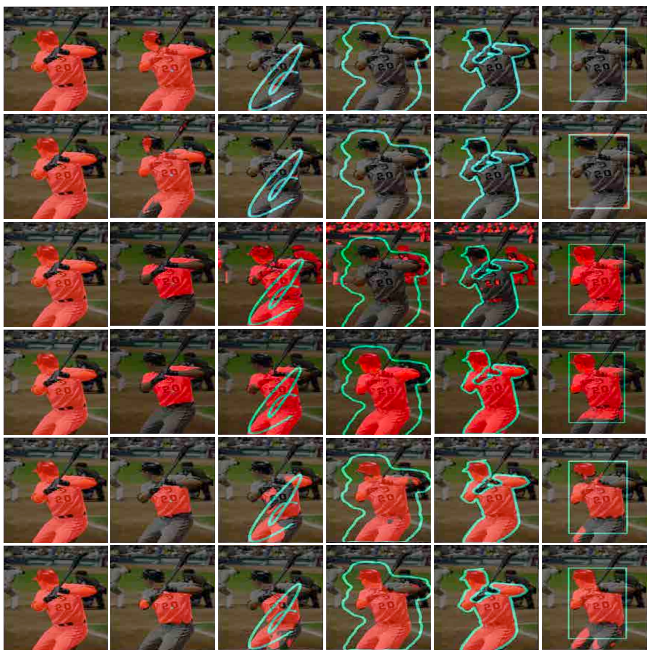



Figure 4. Results for each gesture type for segmentation creation. From the top down: RITM [54] - positive, FocalClick [10] - positive, SAM [25]-C - positive, SAM [25]-R - positive, HRNet-base, HRNet-dataAug. (a) is the input image with ground truth overlaid, (b)-(f) show the results of each method with the gesture used.

during segmentation creation. Intuitively, a tight lasso surrounding the region's boundary provides the most guidance on what to select for interactive segmentation methods while clicks perform the least. Scribbles and rectangles provide similar performance as they may both only envelope *part* of a region of interest. However, we observe that SAM obtains top performance for segmentation creation when using rectangles. This can likely be attributed to rectangles being the supported gesture type that provides the most guidance for SAM during its large-scale training.

We also observe that the disparity in performance between different gesture types is smaller for segmentation refinement (Table 2). For instance, when using *HRNet-dataAug*, the gap between the $\text{RICE}_{\text{local}}$ score achieved by clicks and tight lassos reduces from 24.31 percentage points for segmentation creation to 4.96 percentage points during refinement. One plausible explanation is that the refined segments are typically smaller in size than the entire regions, thereby allowing algorithms to respond more uniformly among gesture types. However, as the spatial size of available corrections diminishes, the utilization of clicks becomes increasingly advantageous. In the NoG setting, described in the Supplemental Materials, we observe that clicks outperform other methods in terms of minimizing failures in reaching a specified IoU. This may be attributed to corrections becoming thinner as they become smaller. Consequently, the effectiveness of boundary level guidance, especially when applied with a fixed thickness (e.g., a radius of 5 in our annotations), may be diminished. In contrast, clicks may be more desirable due to their ability to cover a smaller spatial extent. An additional explanation is that the observed performance advantage of clicks during refinement could be influenced by the implicit bias from training with superpixel previous segmentations in DIG. Future research could explore more efficient methods for generating on-the-fly interactions to alleviate potential biases.

**Analysis with Respect to Segmentation Refinement.** Our results on segmentation refinement reveal that single-gesture methods have limited ability to improve upon previous segmentations, while multi-gesture methods display comparatively better but still suboptimal performance. We observe that context-based methods struggle to enhance previous segmentations when using a single interaction, as evidenced by our proposed RICE metric. One possible explanation is that the widely used metric of IoU may not be a reliable indicator of how well a method has improved a prior segmentation, if at all. For example, when analyzing the $\text{RICE}_{\text{global}}$ score for the RITM method, we observe a relatively low score of 6.11, despite achieving a mean Intersection over Union (mIoU) of 83.74 for the same setting[10]. Moreover, methods that rely on interaction history, such as SAM perform poorly when correcting pre-computed segmentations due to the requirement for knowledge of previous interactions. Under the NoG setting, we find this issue remedied by leveraging subsequent interactions, but find these algorithms struggle when context is not available. Furthermore, SAM-R suffers a disadvantage when refining a previous segmentation as it expects rectangles to *add* content to a segmentation, rather than remove it.

## 7. Conclusion

Our proposed *gesture-agnostic, context-free* interactive segmentation task suports a less cumbersome, more flexible interaction from users. By only accepting user markings on images, it eliminates common additional input requirements, such as the context of an interaction or type of gesture.

**Acknowledgments.** JMD is supported by a NSF GRFP fellowship under Grant No. 1917573 and completed the majority of this work during an internship with Adobe Research. We gratefully thank the participants of our user study, members of the Media Intelligence Lab at Adobe Research for early feedback, and Eric Slyman for helpful discussions about constructing previous segmentations. We also thank the anonymous reviewers for their feedback.

[10]Due to space constraints, we report IoU for each method in the Supplementary Materials.

## References


[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018.

[2] Adobe Inc. Adobe photoshop.

[3] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 392–399, 2014.

[4] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11700–11709, 2019.

[5] Neil Birkbeck, Dana Cobzas, Martin Jagersand, Albert Murtha, and Tibor Kesztyues. An interactive graph cut method for brain tumor segmentation. In *2009 Workshop on applications of computer vision (WACV)*, pages 1–7. IEEE, 2009.

[6] Fredrik Carlsson, Philipp Eisen, Faton Rekathati, and Magnus Sahlgren. Cross-lingual and multilingual clip. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6848–6854, 2022.

[7] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5230–5238, 2017.

[8] Ding-Jie Chen, Hwann-Tzong Chen, and Long-Wen Chang. Toward a unified scheme for fast interactive segmentation. *Journal of Visual Communication and Image Representation*, 55:393–403, 2018.

[9] Xi Chen, Zhiyan Zhao, Feiwu Yu, Yilei Zhang, and Manni Duan. Conditional diffusion for interactive segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7345–7354, 2021.

[10] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. 2022.

[11] Henghui Ding, Scott Cohen, Brian Price, and Xudong Jiang. Phraseclick: toward achieving flexible interactive segmentation by phrase and click. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 417–435. Springer, 2020.

[12] 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 *International Conference on Learning Representations*, 2021.

[13] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, jun 2010.

[14] Marco Forte, Brian Price, Scott Cohen, Ning Xu, and François Pitié. Getting to 99% accuracy in interactive segmentation. *arXiv preprint arXiv:2003.07932*, 2020.

[15] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[16] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[17] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 417–434, Cham, 2020. Springer International Publishing.

[18] Yuying Hao, Yi Liu, Zewu Wu, Lin Han, Yizhou Chen, Guowei Chen, Lutao Chu, Shiyu Tang, Zhiliang Yu, Zeyu Chen, et al. Edgeflow: Achieving practical interactive segmentation with edge-guided flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1551–1560, 2021.

[19] 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.

[20] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. A fully convolutional two-stream fusion network for interactive image segmentation. *Neural Networks*, 109:31–42, 2019.

[21] Suyog Dutt Jain and Kristen Grauman. Predicting sufficient annotation strength for interactive foreground segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1313–1320, 2013.

[22] Shaun K Kane, Jacob O Wobbrock, and Richard E Ladner. Usable gestures for blind people: understanding preference and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 413–422, 2011.

[23] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[24] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019.

[25] 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 preprint arXiv:2304.02643*, 2023.

[26] Navid Alemi Koohbanani, Mostafa Jahanifar, Neda Zamani Tajadin, and Nasir Rajpoot. Nuclick: a deep learning framework for interactive segmentation of microscopic images. *Medical Image Analysis*, 65:101771, 2020.

[27] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov,

Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[28] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *2009 IEEE 12th international conference on computer vision*, pages 277–284. IEEE, 2009.

[29] Xiaokang Li, Mengyun Qiao, Yi Guo, Jin Zhou, Shichong Zhou, Cai Chang, and Yuanyuan Wang. Wdtiseg: One-stage interactive segmentation for breast ultrasound image using weighted distance transform and shape-aware compound loss. *Applied Sciences*, 11(14):6279, 2021.

[30] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM Transactions on Graphics (ToG)*, 23(3):303–308, 2004.

[31] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *2017 IEEE international conference on computer vision (ICCV)*, pages 2746–2754. IEEE Computer Society, 2017.

[32] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 305–314, 2021.

[33] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016.

[34] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[35] Zheng Lin, Zheng-Peng Duan, Zhao Zhang, Chun-Le Guo, and Ming-Ming Cheng. Focuscut: Diving into a focus view in interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2637–2646, 2022.

[36] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13339–13348, 2020.

[37] Zheng Lin, Zhao Zhang, Ling-Hao Han, and Shao-Ping Lu. Multi-mode interactive image segmentation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 905–914, 2022.

[38] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5257–5266, 2019.

[39] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. 2023.

[40] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6172–6181, 2019.

[41] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7086–7096, 2022.

[42] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. *arXiv preprint arXiv:1805.04398*, 2018.

[43] Soumajit Majumder, Ansh Khurana, Abhinav Rai, and Angela Yao. Multi-stage fusion for one-click segmentation. In *DAGM German Conference on Pattern Recognition*, pages 174–187. Springer, 2020.

[44] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625, 2018.

[45] Kevin McGuinness and Noel E O'connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010.

[46] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021.

[47] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8533–8542, 2020.

[48] 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*, pages 724–732, 2016.

[49] 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*, pages 724–732, 2016.

[50] Holger Roth, Ling Zhang, Dong Yang, Fausto Milletari, Ziyue Xu, Xiaosong Wang, and Daguang Xu. Weakly supervised segmentation from extreme points. In *Large-Scale Annotation of Biomedical Data and Expert Label Synthesis and Hardware Aware Learning for Medical Imaging and Computer Assisted Intervention*, pages 42–50. Springer, 2019.

[51] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[52] Jianbing Shen, Yunfan Du, and Xuelong Li. Interactive segmentation using constrained laplacian optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(7):1088–1100, 2014.

[53] 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.

[54] 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.

[55] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2697–2707, 2022.

[56] Guotai Wang, Wenqi Li, Maria A Zuluaga, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE transactions on medical imaging*, 37(7):1562–1573, 2018.

[57] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1559–1572, 2018.

[58] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2254–2258. IEEE, 2021.

[59] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[60] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[61] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. In *28th British Machine Vision Conference, BMVC 2017*. BMVA Press, 2017.

[62] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 373–381, 2016.

[63] Wenxian Yang, Jianfei Cai, Jianmin Zheng, and Jiebo Luo. User-friendly interactive image segmentation through unified combinatorial user inputs. *IEEE Transactions on Image Processing*, 19(9):2470–2479, 2010.

[64] Hongkai Yu, Youjie Zhou, Hui Qian, Min Xian, and Song Wang. Loosecut: Interactive image segmentation with loosely bounded boxes. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3335–3339. IEEE, 2017.

[65] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12234–12244, 2020.