6 Appendices

6.1 Details on data generation

In our work, we compare interactive segmentation with a single click to segmentation conditioned on text saliency. The Phraseclick paper [12] was the first paper to study combining a click and text query for disambiguation. In their experiments on refCOCO, they study different combinations of interactions, and the closest comparison to our work are the experiments they ran with 2 foreground clicks and a single background click. They cite previous work from Xu et al. [36] for their experimental configuration. Negative background clicks are sampled either from other object instances present in a scene, from near the ground truth object boundary, or from anywhere that is not the ground truth object. Positive clicks are sampled subsequently with a minimum distance from each other. Finally, for every object instance in the ground truth dataset, random samples are taken to create augmentations in the training data [36]. The reccomended hyperparameters from Xu et al. include a 40 pixel minimum distance between sampled positive points, and 15 samples per each instance.

To focus on boundary quality of the generated interactive segmentations, we sampled negative points from two of the three strategies proposed by Xu et al. [36]: from other instances of the same class present in a given scene, and from points along the outside boundary of the ground truth points. Additionally, we sampled positive points with 150 minumum distance from each other. We instead took a single sample per instance. This remains a future inquiry to see how the baseline model and ours conditioned on text saliency perform with additional data augmentation. From some anecdotal studies on refCOCO, the performance increase to both the baseline model and ours conditioned on text saliency is roughly 2-3 mIoU. For the fully supervised experiments in Table 5, we take 5 samples per instance for the much smaller dataset refCOCO, and a single sample per instance for COCO.

6.2 Part Disambiguation and Visualizations for COCO

In our main paper, we demonstrate visual examples for the validation set of OpenImages [21]. We also demonstrate in the Experiments section how text saliency improves novel class generalization. Here, we aim to provide examples of how the model trained on OpenImages performs on example images from COCO [23] in order to show model generalization across datasets. As evident in Figure 7, our model qualitatively outperforms the baseline click only model in these settings as well. We show here that conditioning on text saliency also improves the ability of a model to generalize between a whole object and its sub-parts. This is also illustrated in the results shown in Figure 7.

6.3 Experiments in a Fully Supervised Setting

We present results of a fully-supervised version of our text-and-click model and these results are shown in Table 5. The purpose of Table 5 is to understand the delta between the zero-shot segmentation model and a model trained in a fully-supervised manner. Whereas our zero shot segmentation model achieved 66.02 mIoU with a single click and text prompt on refCOCO, our fully supervised model in 5 achieves 68.07 mIoU with a single foreground click and text prompt, and 72.89 with two foreground clicks, one background click and a text prompt.

(a) RGB Input + Click (b) Baseline Prediction (c) Our Prediction for 'Shirt' and 'Hat'

(d) RGB Input + Click (e) Baseline Prediction (f) Our Prediction for 'Tie' and 'Person'

Figure 7: A comparison of the click-only baseline to text-saliency segmentation on the task of part disambiguation. The model here was trained in a zero-shot manner on OpenImages with 64 seen classes, and evaluated on validation images from COCO. The categories chosen are from the unseen class set. Text saliency conditioning helps the segmentation model disambiguate subparts such as the "tie" from the overall object of "person." Similarly, the segmentation model conditioned on text saliency is able to differentiate the classes of shirt, hat and person in the top row.

This demonstrates that constraining the model to a limited set of classes does not lead to a significant performance drops, we attribute this to our text saliency conditioning. The baseline zero shot segmentation refCOCO model with only a single foreground click achieved 62.99 mIoU, indicating a more notable performance drop versus the model conditioned on text saliency. Similar results are seen for COCO in Figure 5 where the model trained in fully supervised mode achieved 47.17 mIoU for a single foreground click and text prompt; our zero shot model trained on only 20 seen classes achieves only 38.42.

Table 5 additionally contains an ablation experiment which ablates the number of interaction signals received by the model and determines how performance varies. We establish that inputting a text prompt counts as an interaction, and we therefore compare various configurations of text prompt inputs, foreground and background clicks. We hypothesized that additional interactions would decrease the utility of text saliency, because the object or subpart to segment would already be clearly defined. We find this generally confirmed, especially for results on the COCO dataset. We can see this when comparing rows 1-2 and rows 3-4 in Figure 5: under the condition of a single foreground click, the addition of a text prompt boosts mIoU by 10.35 for COCO; whereas under the condition of 2 foreground clicks and a single background click, the addition of a text prompt only boosts mIoU by 2.19.

6.4 Comparison with SAM Model

In our experiments, we compared our model to the Segment Anything Model (SAM) from Meta AI. Performing comparisons on the zero-shot segmentation setting was infeasible due to the extremely large number of GPUs required to retrain. To do so would require retraining

Interaction			Overall mIoU	
Text Input	Pclicks	Nclicks	COCO	refCOCO
✓	2	1	**54.46**	**72.89**
	2	1	52.27	71.53
✓	1	0	**47.17**	**68.07**
	1	0	36.82	66.16
	1	1	47.13	66.23

Table 5: Results of our model in the fully supervised setting over the COCO and refCOCO datasets for the text-click instance segmentation task. We convert text input to a heatmap using Maskclip. The left hand side of the table shows the number of inputs given to the model in terms of text-saliency heatmaps, positive clicks (PClicks) and negative clicks (NClicks). The interaction setting with the highest mIoU is bolded for reference.

the SAM model on a limited number of seen classes in its dataset. To the best of our knowledge, data category labels were not available at training time. In lieu of these experiments, we conduct comparisons to the pre-trained SAM network.

In Table 3, we explained that SAM outputs multiple mask proposals. We explore multiple strategies to filter their mask proposals to the best available proposal. We previously discussed using the CLIP similarity of each mask proposal crop to the ground truth text prompt. This was meant to produce an even comparison, since our model has access to the ground truth category label of an instance to generate the text saliency map. We also discussed the SAM confidence score. For sake of thoroughness, we also re-implement the Oracle score described in the SAM paper. They note that their model can be penalized by automated evaluation metrics because it suggests multiple masks; and note that the model produces SOTA results if allowed to compare its mask proposals to the ground truth one. See the Experiment sectionin the main paper for details. This suggests that the SAM model struggles with disambiguating mask proposals, though it can often suggest high quality ones.

6.5 Looking at Distractors and Neighboring Object Segmentation

We analyze the role that distracting objects play in generating interactive segmentation. A given image can have multiple classes present, for example a table and a lamp. For a given class, multiple instances can be present, for example, a parking lot with multiple instances of the class ‘vehicle’. Since the model learns to segment guided by a click and an open-vocabulary salience map (generated from a text category). This becomes a more challenging task the more objects and instances that are present, particularly the closer they are in proximity. We achieve the best results on refCOCO, a subset of COCO data re-sampled to make human annotation easier.

Results area available in Figure 8. In this experiment, we measure the number of instances of the same class present in a given image, and record the mIoU for each instance in the validation set along with the number of distractors present, only for instances of unseen classes. Our model consistently outperforms the baseline model, though the gap is similar across the number of distracting objcets present.

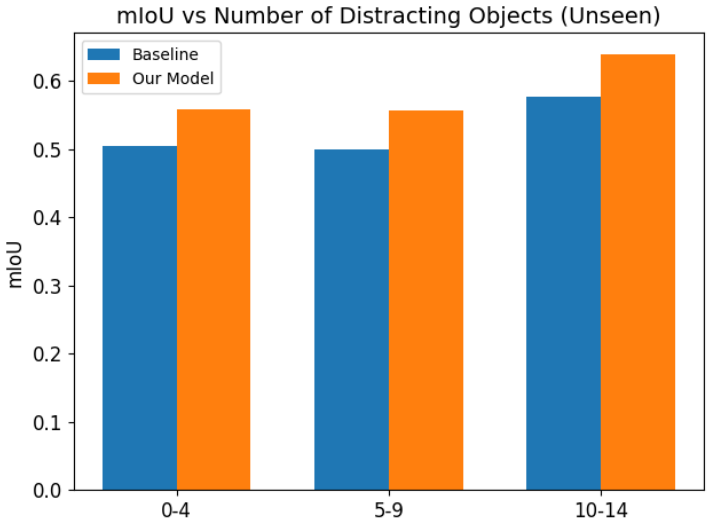

Figure 8: Bar chart displaying the effect of number of objects on segmentation quality. Using the OpenImages dataset we plot the average mIoU for images with N number of objects of the same class present. This analysis is only for instances of classes not seen during training. The model is trained on OpenImages with 64 classes as seen, and evaluated on the OpenImages validation set with all classes available. The baseline here is the click-only model compare to ours conditioned on text saliency. Our model consistently outperforms the baseline model, though the gap is similar across the number of distracting objcets present.

6.6 Limitations and Future Work

We identify two main failure modes of the proposed model. The first instance results as a cascading error in cases with a low quality heatmap. In our experiments, we tried a few saliency techniques including GradCAM [29], Chefer 2021 et al. [3] and MaskCLIP [40]. We found MaskCLIP to qualitatively perform the best, but improved saliency maps remains an important future line of inquiry. Sometimes, the heatmap helps to localize a given text query, but the segmentation network we train still fails to accurately segment it. We can see this second failure mode illustrated in Figure 9. In the example in the bottom row of the figure, the heatmap has reasonably high probability over the pixels of the car wheel however the predicted segmentation contains the pixels of the entire car. Similarly, in the example in top row - containing an instance segmentation for a hat - the heatmap for the hat is high quality, but it predicts part of his whole body. We suspect that this is due to an imbalance of annotations in the training data; there are plenty of instances of whole objects such as automobiles or entire person silhouettes, but very few of a wheel, license plate, or hat.

6.7 Comparison to Interactive Click Methods

In Segment Anything [20] Sec 7.1, Krillov et al. compare SAM to other interactive segmentation baselines (RITM, SimpleClick, and FocalClick) on single-click segmentation across 23 datasets. In Figure 9c and 9d of[20], SAM significantly outperforms all other methods on a single click, though the gap is much smaller for 2,3, or 5 clicks. This is because many interactive segmentation models are trained for mask refinement as opposed to generating the optimal proposal from a single click. These papers report the number of clicks to achieve

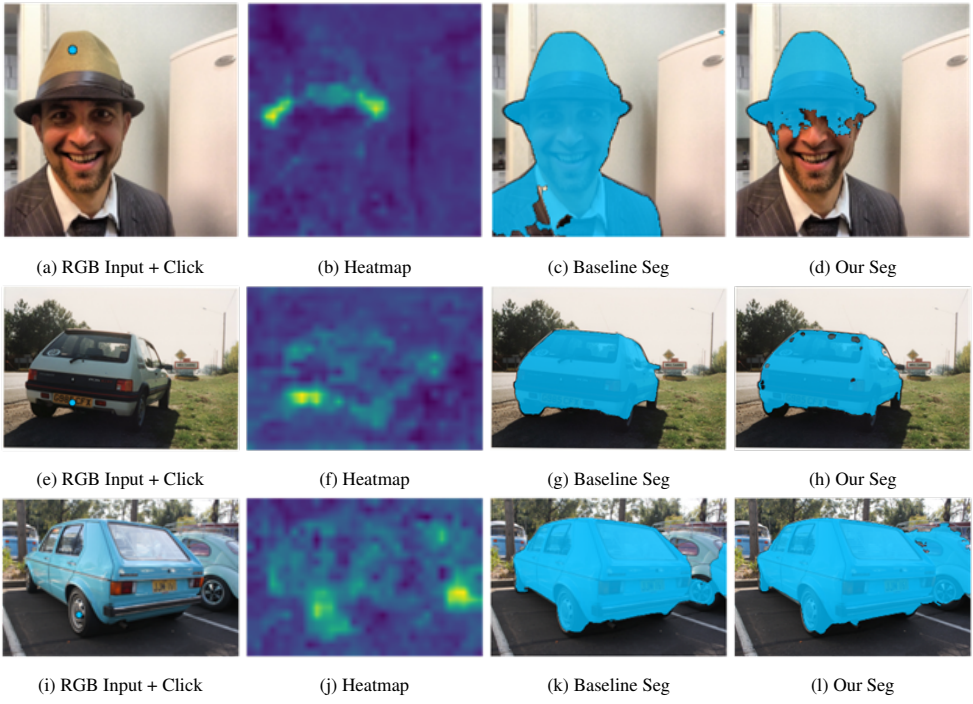

Figure 9: Examples of failure cases of text-click instance segmentation of our model. These examples show instances where despite the text saliency localizing the object of interest, the segmentation mask fails. We observe this largely happens in overlapping objects or when the queried category is a sub-part of a larger object. Categories: (a) "Cowboy hat", (b) "Vehicle registration plate", (c) "Wheel".

a target IoU, but we are interested in minimizing interactions by combining text and clicks. We compare to RITM [31] in Tab. 6, and show better generalization on COCO when trained on 20 VOC classes to unseen COCO classes. RITM is stronger on OpenImages, but the gap between seen and unseen is larger, suggesting RITM is a stronger click baseline but generalization could benefit from text saliency conditioning.

6.8 Boundary IoU Metrics

Please see table 7. As you can see, using boundary IoU instead of mIoU, we achieve similar results, where our model slightly beats SAM on COCO validation.z

6.9 Comparison to Referring Expression Segmentation Methods

Our method is able to generalize to unseen classes with text input by using pre-trained CLIP. We show this in the paper with our “unseen” metrics. PhraseClick, VLT and LAVT have no mechanism to do this and do no evaluate on unseen classes.

The largest difference between our model, LAVT and VLT is that we can segment completely unseen classes at test time. We achieve 33.45 mIOU on 60 COCO unseen classes while training on only 20 seen classes. We unlock this capability by training on saliency

Model	Train Data	Eval Data	mIoU		
			Overall	Seen	Unseen
RITM [31]	SBD	COCO Validation	38.86	45.00	30.33
Ours	COCO[voc classes]	COCO Validation	38.42	42.06	33.45
RITM [31]	SBD	OpenImages	49.42	67.39	47.17
Ours	COCO[voc classes]	OpenImages	44.55	53.16	41.87

Table 6: Comparing generalization of models to unseen classes. Comparison of RITM interactive segmentation model with a single click and our model with a click + text label. Using RITM checkpoint trained on all VOC classes with the SBD data. Our model was trained on VOC classes of COCO. COCO validation has 80 classes, and OpenImages has 300 classes. We will include a comparison while training on SBD in the camera ready.

Dataset	Model	mIoU		
		Overall	Seen	Unseen
COCO	Ours	39.62	40.51	38.47
COCO	SAM [20]	38.93	37.63	40.65

Table 7: Boundary IoU comparisons on MS COCO.

maps from Maskclip that is able to leverage all of the knowledge of a pre-trained CLIP model. We compare with LAVT and LVT in fully supervised setting (all classes are seen) in Table 8 and show that we are able to match their performance with 3 clicks. LAVT and VLT have not published numbers from unseen classes.

Also LAVT and VLT [11] require more specific language than our model ("guy in black sitting to left leaned over" (Fig 6 [37]) vs. ours - "person"). We achieve similar performance with less specific text supervision. This is important since annotating referring expression datasets at scale is expensive. In contrast, our method only requires the much more readily available, ground-truth class.

6.10 Comparison to Phraseclick

The PhraseClick [12] was published before Vision-Language joint pretraining became a common method. Therefore, [12] propose an attention attribute mechanism, whereby the visual features are global average pooled into the same common dimension as the embedding dimension for the text representation. Text input is processed using word2vec and a trainable bi-LSTM. The text input is not initially aligned with the distribution of visual features. At inference time, if a novel query is presented, the PhraseClick model will be unable to use the text information to make an improved segmentation.

In our work meanwhile, we use the Maskclip technique to produce a spatial saliency map, for any possible novel text query, that provides a rough guess for the location of a given query. Maskclip retains the explicit spatial information providing a useful initial guess to the location of an object. Our model is trained in a class agnostic manner after extracting a

Model	Input	mIOU
LAVT	Phrase	72.73
VLT	Phrase	65.65
Ours	Class Name + 3 clicks	72.89
Ours	Class Name + 1 click	68.07

Table 8: Comparison of our method to Referring expression segmentation algorithms. For 3 clicks we sample 2 foreground and 1 background click. For 1 click we sample 1 foreground click. All models were trained on RefCOCO.

heatmap guess, and so learns to segment any given prompt.

PhraseClick [12] did not release code nor model weights, we cannot provide visual comparisons.