conSAMmé: Achieving Consistent Segmentations with SAM

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

Abstract

Multi-output interactive segmentation methods generate multiple binary masks when given user guidance, such as clicks. However, it is unpredictable whether the order of the masks will match or whether those masks will be the same when given slightly different user guidance. To address these issues, we propose conSAMmé, a contrastive learning framework that conditions on explicit hierarchical semantics and leverages weakly supervised part segmentation data and a novel episodic click sampling strategy. Evaluation of conSAMmé's performance, click robustness, and mask ordering show substantial improvements to baselines with less than 1% extra training data compared to the amount of data used for the baseline.

1. Introduction

Computational photography is moving beyond 'simple' point-and-shoot devices [13, 23, 25, 26], with a central challenge for capturing and manipulating digital images being to isolate regions of interest by classifying which pixels belong to the target region (i.e., image segmentation). For instance, one may want to locate a person in a portrait [11, 38] or interactively remove shadows [7, 8]. With modern fully automated methods [3, 18, 35, 40] still performing poorly at times, interactive segmentation methods are a promising way for infusing a small amount of user guidance, usually in the form of clicks [2, 15, 21, 36], to efficiently acquire high quality segmentations. While early interactive approaches generated a single output for a given user input [2, 32, 36], the trend has instead shifted to generating multiple segmentation options given the inherent ambiguity of such input [15, 20]. However, for such multi-output methods, two key issues arise.

The first limitation of existing methods is that the order of the resulting masks can be inconsistent. As exemplified in **Figure 1(a)**, a single click results in the entire object appearing in the first mask position and its part mask in the second position, while another click reverses that order. As a result, a user must hunt for the correct mask instead of knowing where to find the object or part a priori. We suspect this happens because the model does not learn explicit positions for each hierarchical level (e.g., "first mask = object" and "second mask = part"). Existing work [15, 20] instead ignores ordering based on semantics when picking for training a mask that best matches the ground truth.

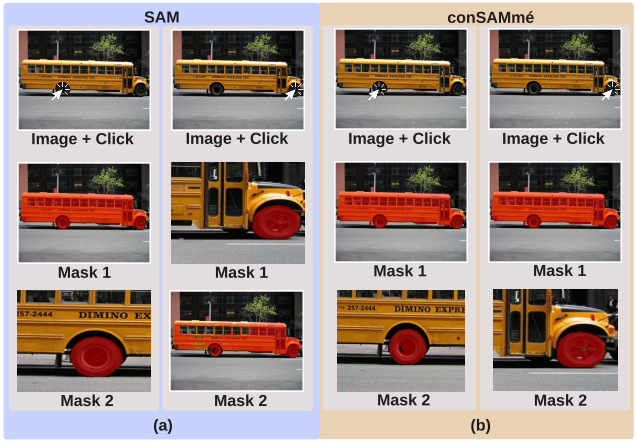

Figure 1. Exemplar image with a user-input click (top row). In panel (a), SAM [15] produces an inconsistent ordering of segmented regions, for example placing the object mask in the first position for one click (left) and in the second position for another (right). Moreover, even though both clicks target the same object, clicking on the front yields a higher-quality object mask than clicking on the rear tire. In contrast, panel (b) shows how a model trained with our conSAMmé framework not only produces consistently accurate object-level masks from both clicks but also maintains a stable ordering (object in position 1, part in position 2). Parts and zoomed and cropped for ease of visualization.

Second, these algorithms can produce different object-level masks depending on the click location. **Figure 1(a)** illustrates how clicking on the front tire of a bus yields a poorer mask than clicking on its rear tire, forcing the user to guess the best click location or necessitate further refinements. This challenge likely stems from the fact that methods assume an entity's appearance (e.g., size, aspect ratio) when deciding how to segment, which can fail when objects appear in varied poses or views [22]. We hypothesize that learning part-whole relationships (i.e., hierarchical semantics) would be more robust: for instance, a click on a tire

of a bus (a part) or its head should lead the model to also include the entire bus (the object), as shown in **Figure 1(b)**. By learning how parts fit into wholes, the model can leverage knowledge from multiple semantic levels rather than relying on a single global property (i.e., appearance).

To overcome these limitations of prior work, we propose the following contributions. *First*, we introduce a contrastive learning framework, **conSAMmé** (**C**onsistent **S**egment **A**nything **M**odel for **M**ulti-level **E**xtraction), for teaching the model to generate hierarchically-consistent output segmentations, which consistently positions objects as the first output and parts[1] as the second output. This is exemplified in **Figure 1(b)**. *Second*, we propose a novel click sampling strategy based on object-part relations to facilitate the learning of part-whole relationships to generate consistent results. *Third*, to account for the relative scarcity of part segmentation training data, and to preserve zero-shot performance, we develop a weakly-supervised data generation technique using SAM's [15] automatic mask generation and object-level ground truth, that partitions an object into parts to facilitate training. Experiments show conSAMmé outperforms baselines at generating consistent segmentation outputs while only requiring less than 1% of additional training data compared to SAM.

This work can directly benefit computational photography, both for enhancing and restoring images with less human involvement, such as when adjusting lighting on a face, changing the color of a garment sleeve, or modifying the saturation of an entire object. We also anticipate this work can benefit a broader audience. It can accelerate creating datasets by reducing the amount of required human intervention, including for creating part-level [1, 9, 30] and subpart-level [28] segmentation datasets. It could also extend beyond interactive segmentation. For instance, language models that generate multiple completions for a user to choose from [4, 6, 37] could incorporate entity consistency when deciding what information to provide to a user, such as concise explanations for experts and detailed explanations for lay users.

2. Related Work

Multi-Output Interactive Segmentation Algorithms. Numerous models generate multiple segmentations from a single user interaction (e.g., a click). The focus for early approaches, such as [19], was to generate multiple diverse segmentations as a precursor to establishing a single, high-quality segmentation. MultiSeg [20] was the first method centered on generating multiple segmentations as the target output by producing many segmentations corresponding to aspect ratios and computing loss on the top-k best matches with the ground truth, where k is a hyperparameter. Extending this paradigm, the Segment Anything Model (SAM) [15] and its variants [12, 21, 34] generate multiple segmentation masks such that they have spatial relationships where each mask either *contains* another segmentation or is *contained* within another segmentation. As discussed in the Section 1, our work extends this latter work by introducing a method that can generate consistent object-level masks regardless of where the user clicks.

Weakly-Supervised Interactive Segmentation. Weak supervision has been used for both single-output and multi-output interactive segmentation methods in order to reduce reliance on costly, dense pixel-level annotations.[2] For instance, MultiSeg [20] uses neighboring objects, such as a person riding a horse, to train scale-aware segmentation, achieving more diverse outputs. SAM [15] adopts a model-in-the-loop approach with iterative human refinements, producing object, part, subpart, and background masks without explicit semantic labels. Extensions of SAM emphasize uncertainty quantification in pseudolabels for training [21], or similar weakly supervised strategies for video segmentation [31], which are orthogonal to our task, as they either leverage epistemic uncertainty to refine existing segmentations or focus on segmenting regions in videos—both typically relying on a single model output. In contrast, our work focuses on enhancing consistency across multiple segmentation outputs. FocalClick [2] and DIG [27] use superpixel degradations to generate imperfect masks such that models can learn to perform corrections. However, these weak supervision strategies usually emphasize objects without incorporating hierarchical labels (e.g., object and part). As a result, they cannot enforce that different clicks on the same object should yield the same object-level mask. **Figure 1(a)** illustrates this limitation: because the training data does not model part-whole relationships, clicking on different parts of the same object produces inconsistent object-level masks. To incorporate hierarchical relationships into training, we propose converting existing object segmentation datasets into part-level annotations. We leverage SAM's automatic mask generation, which partitions each image into meaningful regions, and then treat these regions that fall within an object's ground truth mask as its parts. This approach provides weakly supervised part annotations that preserve part-whole relationships. This strategy can enable multi-output interactive segmentation models to generate semantically consistent outputs, such as consistently placing the best object segmentation in the first mask and its part in

[1] While our method could be extended to subpart-level hierarchies, we limit our scope to part-whole segmentation due to the lack of instance segmentation datasets with subpart annotations at the time of writing.

[2] Also related are open-vocabulary segmentation methods which rely on weak supervision [10, 17, 33]. However, to our knowledge, none of these methods address our task of generating consistent segmentation outputs in a predictable hierarchical structure with objects and parts in interactive, click-based scenarios.

the second, while also improving robustness to initial click placement.

3. Method

We now introduce our **conSAMmé** framework, which incorporates several design innovations to the Segment Anything Model (SAM) [15], as summarized in **Figure 2**. In what follows, we describe the model architecture, training approach, and training data generation process.

3.1. Background: SAM

Our proposed framework leverages SAM's powerful architecture, which operates in two distinct modes: single-mask mode where a single segmentation mask is produced and multi-mask mode where three masks are generated to reflect potential ambiguities inherent in user prompts. It consists of three key modules: a Vision Transformer [5] (ViT) to encode the image, another set of parameters to encode the input geometric prompts (e.g., clicks and rectangles), and a transformer decoder to decode the resulting features into segmentation masks by using bi-directional attention between prompt tokens and image embeddings. More specifically, each mask is generated by taking the dot product between a vector derived from the token-to-image attention outputs and the upsampled image-to-token attention features. Formally, the mask multidimensional array, $\mathcal{M}$, is constructed as:

$$\mathcal{M} = \sigma(\mathcal{F}\mathcal{T}^{\top}), \tag{1}$$

where σ denotes the sigmoid function, $\mathcal{F} \in \mathbb{R}^{(H \cdot W) \times 32}$ represents the upsampled image-to-token features (with H and W being the spatial dimensions of the image), and $\mathcal{T} \in \mathbb{R}^{4 \times 32}$ is the MLP outputs from the token-to-image attention layers. The authors of SAM refer to each of these rows as output mask tokens as they are responsible for combining the feature of $\mathcal{M}$ to form a coherent mask. After computing $\mathcal{M}$, it is reshaped into a multidimensional array with dimensions $H \times W \times 4$ and then thresholded to obtain the final binary masks. In Section 3.2, we elaborate on how we leverage the row vectors in $\mathcal{T}$ to enhance consistency during training.

3.2. conSAMmé Framework

The core contributions of our work is its training approach, which is designed to enforce consistent mask ordering by incorporating contrastive learning and utilizing an episodic click sampling strategy.

Mask Ordering Guidance. We introduce a loss function to enforce an explicit ordering for masks output by the model. Specifically, we supervise the first mask output, denoted as $\hat{m}_1$, to be an object-level segmentation and the second mask output, $\hat{m}_2$, to be a part-level segmentation. We compute the loss as a weighted sum of focal and dice losses, as done in SAM [15], formally expressed as:

$$\mathcal{L}_{sam} = \sum_{(\hat{y},y)\in\Omega} \left[20 \cdot Focal(\hat{y}, y) + DICE(\hat{y}, y)\right], \tag{2}$$

where $\Omega = \{(\hat{m}_1, m_o), (\hat{m}_2, m_p)\}$; $\hat{m}_1, \hat{m}_2 \in [0,1]^{H \times W}$ represents the predicted masks for the object and its parts respectively; and m_o, $m_p \in \{0,1\}^{H \times W}$ denotes the ground-truth masks for the object and its parts respectively. This formulation not only enforces the desired *hierarchical ordering*, with object masks always supervised in the first position and part masks in the second, but also reinforces that different parts of an object should share the same object-level mask. This standardized evaluation encourages reproducible results, which random clicks do not provide.

Click Sampling. We further encourage the model to learn different parts should share the same object mask with an episodic click sampling strategy. This approach contrasts the status quo [2, 15, 32, 36] of generating synthetic clicks by sampling points randomly within an object during training. For any object with n parts, we generate an episode (a set of user clicks for both the object and its parts) of positive clicks indicating content we wish to segment, denoted by $\mathcal{E}_c = \{c_1^+, c_1^+, \ldots, c_n^+\}$, with each click corresponding to a different part. During training, these clicks are sampled randomly from the ground-truth regions of each part. Then, aligning with the status quo of interactive segmentation [32, 34, 36], evaluation relies on using the center points of the objects.

Part-Object Contrastive Learning. To enhance the model's ability to capture the relationships between objects and their constituent parts, we introduce a contrastive loss on the token-to-image attention vectors produced by SAM's mask decoder (i.e., the matrix $\mathcal{T}$ in Equation 1). Our intuition is that clicks corresponding to different parts of the same object should yield similar object-level representations while exhibiting distinct part-level representations.

Concretely, consider an image for which we have an episodic set of clicks, $\mathcal{E}_c$. For each click $c_i \in \mathcal{E}_c$ (alongside an input image), the mask decoder produces a corresponding token-to-image attention output:

$$\mathcal{T}_i \in \mathbb{R}^{K \times d}, \tag{3}$$

where K is the number of token outputs ($K = 4$ in SAM) and d is their dimensionality (with $d = 32$ in our experiments). We assume the second[3] row of $\mathcal{T}_i$ represents the object-level feature and a designated subsequent row (i.e.,

[3]The first row of $\mathcal{T}$ corresponds to the single mask mode in SAM.

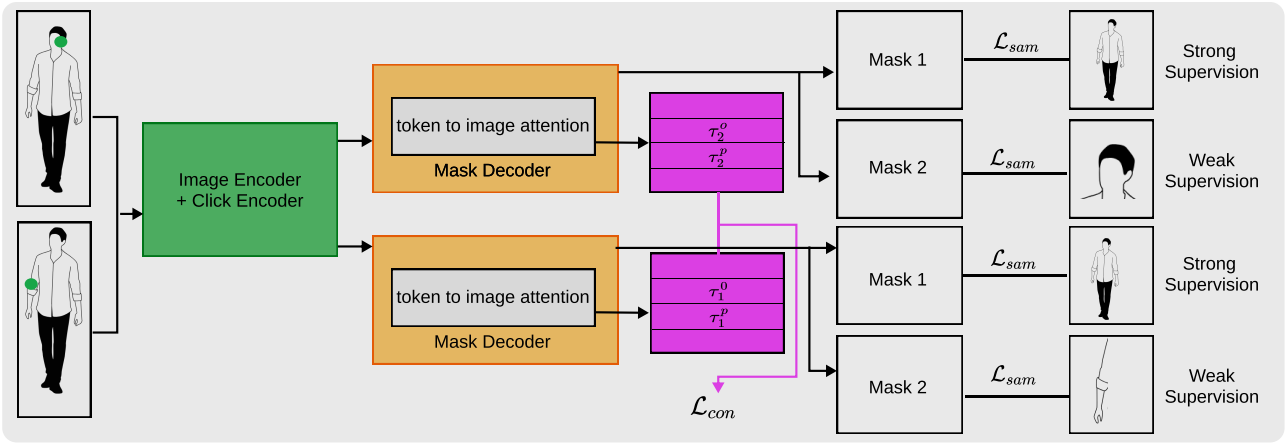

Figure 2. Overview of our proposed conSAMmé framework. Given an episode of clicks on an image, such as a click on the head and arm of a person as shown above, we encode the image and click and feed each of those features into a mask decoder. We then perform contrastive learning on the mask tokens produced from the token to image attention MLP layers. For the masks produced by each of the clicks, we supervise the first position with the object ground truth and the second with a weakly supervised part mask.

the third row) represents the part-level feature. We denote these as:

$$\tau_i^o \in \mathbb{R}^d \quad \text{and} \quad \tau_i^p \in \mathbb{R}^d. \tag{4}$$

For any two distinct clicks c_i and c_j (with $i \neq j$) on different parts of the same object, we aim for the similarity between their object-level features to be high while the similarity between their part-level features to be relatively lower. To enforce this, we define a contrastive loss for the pair (i, j) as:

$$\ell_{con}(i,j) = \max\Big(\mu - \Big[\underbrace{\text{sim}\left(\tau_i^o, \tau_j^o\right)}_{\text{want: large}} - \underbrace{\text{sim}\left(\tau_i^p, \tau_j^p\right)}_{\text{want: small}}\Big], 0\Big), \tag{5}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and μ is a margin hyperparameter (set to 0.5 in our experiments). We subtract part similarity from object similarity to directly optimize for greater intra-object consistency and greater inter-part distinctiveness. The margin ($\mu = 0.5$) was chosen to reflect a balanced threshold between over-clustering and under-separating representations and was selected based on early experiments. Averaging over all distinct pairs yields the overall contrastive loss:

$$\mathcal{L}_{con} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \ell_{con}(i,j), \tag{6}$$

with

$$\mathcal{P} = \{(i,j) \mid i,j \in \{1,2,\ldots,n\},\, i \neq j\}. \tag{7}$$

The total loss optimized during training is then a weighted sum of the segmentation loss $\mathcal{L}_{sam}$ and the contrastive loss:

$$\mathcal{L}_{total} = \mathcal{L}_{sam} + \lambda\, \mathcal{L}_{con}. \tag{8}$$

We follow prior work that weights auxiliary losses less heavily than the primary segmentation loss [39] and set λ to 0.5 in our experiments.

The hyperparameter values for μ and λ were selected to balance contrastive and segmentation objectives and were found to perform well across datasets. We briefly explored variations during the early stages and found results were not highly sensitive to small changes.

3.3. Training Data Generation

Part segmentation data is needed both for out-of-distribution generalization evaluation and training data. Given the scarcity of part segmentation datasets, we adopt a weakly supervised approach to generate such data: it leverages existing object segmentation datasets to derive part-level annotations via an automated process, thereby bypassing the need for costly manual labeling. It is inspired by SAM's granularity-agnostic, model-in-the-loop training approach, where SAM iteratively refines its own predictions. We instead extend this framework to facilitate the learning of *granularity-ordered* outputs.

At a high level, our approach operates as follows. Given an image and its associated object segmentation(s), we first employ SAM's automatic mask generation[4] (AMG) capability (using a grid size of 32) to produce candidate segmentations for all relevant regions of the image. For each object ground truth, we then derive weak part annotations by performing an element-wise multiplication of the binarized object mask with the corresponding AMG mask. This operation isolates regions within the object that are likely to correspond to distinct parts. To facilitate our episodic train-

[4]We use the most powerful variant of SAM, ViT-H [5] for generation to maximize quality.

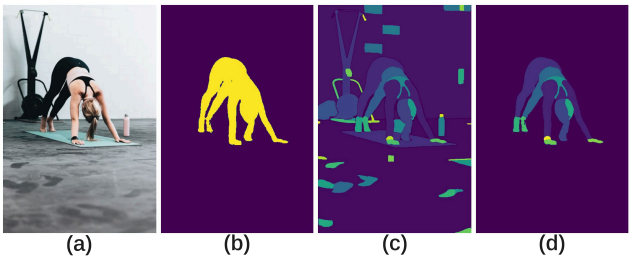

Figure 3. Example of our proposed method for generating weakly-supervised part segmentation data. Given an image (a) and its corresponding object mask (b), we use SAM's automatic mask generator to generate region segmentations (c). We then element-wise multiply the binary object mask with the region segmentation mask to generate part segmentations (d).

ing framework receiving a robust supervisory signal, which requires multiple parts per object, we discard candidate part segmentations lacking at least two distinct parts. A high-level overview of our approach is provided in Algorithm 1, with an illustration in **Figure 3**.

Algorithm 1 Weakly Supervised Part Segmentation Training Data Generation

masks $\leftarrow$ [ ] *// Initialize empty list*
$M_{\text{AMG}} \leftarrow$ gen_AMG_mask(I) *// Generate AMG mask from image*
for m_o **in** $\{m_o\}$ **do**
  $m_p \leftarrow m_o \odot M_{\text{AMG}}$ *// Compute part segmentation*
  if num_parts(m_p) > 2 **then**
    masks.append(m_p) *// Add valid mask*
  end if
end for
return masks *// Return final masks list*

4. Experiments

We now assess the performance of our proposed conSAMmé framework and ablate specific design choices to highlight their importance.

4.1. Experimental Design

We evaluate our framework by simulating a single positive click placed on a given part. Traditional interactive segmentation experiments are designed to iteratively correct model mistakes. However, in our multi-output setting, that is less applicable as we focus on generating the correct set and ordering of outputs, rather than optimizing for the single most correct output. From a single click on the middle of a part, we measure models' abilities to correctly output an object in the first output, and the desired part in the second output.

Baselines. We adopt the Segment Anything Model (SAM) as our baseline since our framework extends it[5]. We also adopt another variant of SAM, HQ-SAM [12], which is a high-quality adaptation that improves mask accuracy by incorporating a fine-grained segmentation head and leveraging high-resolution features. We use the ViT-B [5] versions of both baselines. To ensure a fair comparison, we sort the baseline's multimask outputs in descending order by size (i.e., number of mask pixels). The underlying intuition is that the object-level mask typically covers the largest amount, while a part-level mask is expected to occupy the second largest amount. While MultiSeg [20] could be another valuable baseline, its code is not publicly available.

Implementation Details. We implement our conSAMmé framework using the ViT-B version of SAM. We train conSAMmé for 10 epochs with a batch size of 8 episodes and set the maximum number of masks in an episode to 32. We use the AdamW [24] optimizer with $\beta_1 = 0.9, \beta_2 = 0.99$ with a learning rate of $1\text{e}-5$. We fully fine-tune the mask decoder and freeze the prompt encoder. For the image encoder, we employ a layer-wise learning rate strategy, adopted from [21]. This learning rate scheduling strategy applies a layer-wise decay approach, where the learning rate decreases by a factor of 0.8 for each successive layer of the image encoder, starting from the initial learning rate. Layers closer to the input receive smaller learning rates, encouraging stable feature extraction, while higher layers and non-encoder components (e.g., heads and necks) retain higher learning rates for faster adaptation. Additionally, weight decay is applied selectively, with no decay applied to normalization and bias parameters. To perform model selection, we retain the model with the best validation loss. We perform training on 8 NVIDIA A100 GPUs.

Datasets. For training, we leverage the training sets of EntitySeg [29] and HQ-Seg [12] because they complement each other: EntitySeg features 33,227 scenes with diverse object categories at varying scales, while HQ-Seg contains 44,320 images with salient regions, a common scenario in interactive segmentation, such as portrait photography [14]. In our experiments, we randomly allocate 80% of each dataset for training our conSAMmé framework and reserve the remaining 20% for validation. Of note, this additional data represents only $\sim$0.7% additional training data compared to the training data used for SAM.

For evaluation, we test models in a zero-shot setting on three popular part segmentation benchmarks: Pascal Part [1] (which provides annotations across scenes with

[5]While SAM2 [31] *could* be adapted for our task, we build on SAM due to its lower computational overhead and comparable performance on static images.

Method	Pascal Part [1]			PartImageNet [9]			PACO [30]		
	$\text{mIoU}_{\text{Obj}} \uparrow$	$\text{mIoU}_{\text{Part}} \uparrow$	CVR@0.5 $\downarrow$	$\text{mIoU}_{\text{Obj}} \uparrow$	$\text{mIoU}_{\text{Part}} \uparrow$	CVR@0.5 $\downarrow$	$\text{mIoU}_{\text{Obj}} \uparrow$	$\text{mIoU}_{\text{Part}} \uparrow$	CVR@0.5 $\downarrow$
SAM [15]	69.73	25.67	45.83	64.05	37.78	35.44	20.50	25.11	44.19
HQ-SAM [12]	70.02	24.48	44.32	64.56	36.72	33.38	20.02	23.63	43.56
conSAMmé	**71.95**	**36.11**	**32.21**	**76.27**	**42.99**	**17.43**	**21.52**	**35.92**	**40.56**

Table 1. Experimental results of SAM vs. conSAMmé on three popular part segmentation datasets. Overall, conSAMmé outperforms SAM with respect to both segmentation quality and mask consistency. All results are zero-shot.

single and multiple objects), PartImageNet [9] (which offers part segmentations for 11 objects categories), and PACO [30] (which features high-quality annotations with a long tail of categories). We use the validation sets of Pascal Part and PACO, and the test set of PartImageNet, consisting of 4772, 9443, and 4598 images, respectively. This diverse evaluation suite enables us to assess the generalization of models across varying segmentation challenges.

Evaluation Metrics. To evaluate segmentation accuracy, we compute the mean Intersection-over-Union (mIoU) for both object- and part-level predictions overall all samples, where a click is placed at the center of a part and the object prediction comes from the first mask and the part comes from the second in a multi-output setup. IoU scores range from 0 (no overlap) to 1 (perfect overlap).

In addition, we introduce the Consistency Violation Rate (CVR) to quantify the consistency of the model's hierarchical semantics (i.e., does the model recognize that all parts belong to the same object?). In other words, we assess the model's ability to consistently group distinct parts as belonging to the same object. Moreover, this metric measures how robust a model is to the initial click by checking if similar inputs provide similar outputs. For an object with n parts, we simulate a click on each part and collect the corresponding object-level mask predictions, forming a set M. After thresholding these masks at 0.5, we compute the pairwise IoU between every two masks in M. If the IoU between a pair is below a threshold δ, we count that as a consistency violation. The CVR is then defined as the fraction of such violations over all possible mask pairs:

$$\text{CVR}(M) = \frac{1}{\binom{|M|}{2}} \sum_{i<j} \mathbb{1}\Big[\text{IoU}\left(M[i], M[j]\right) < \delta\Big], \quad (9)$$

where $\mathbb{1}$ is the indicator function that returns 1 when the IoU is less than δ and 0 otherwise, and $M[i]$ denotes the mask corresponding to the i^{th} part. We set $\delta = 0.5$ to balance precision and recall, in line with common practices in evaluation metrics such as mean average precision (mAP). Values range from 0 (all parts have a similar object mask) to 1 (all parts have no similar object masks).

4.2. Results

Results across the three datasets are shown in **Table 1**. We now analyze these results with respect to segmentation quality, accuracy, and mask ordering.

Results with respect to segmentation quality. For segmentation performance, we first notice that conSAMmé outperforms SAM across all granularities. Interestingly, the largest gains appear in part-level segmentation, even though we allocate more supervision to objects (if an object has n parts, we apply the loss to the same object ground truth n times). A plausible explanation for this paradox is that SAM was already strong at object-level segmentation, whereas conSAMmé adds explicit part distinctions via contrastive loss and our episodic click strategy. Repeatedly training on the same object through different part clicks forces the model to learn a consistent and finely detailed object boundary, because it sees multiple *views* (i.e., results from different part clicks) of the object in each episode. In tandem, the part–object contrastive loss ensures that different parts of the same object learn distinct feature compositions, leading to finer intra-object delineations. Consequently, while object masks also benefit from the enhanced supervision, part segmentation sees especially pronounced improvements due to the explicit supervision.

When examining results per dataset, we observe segmentation performance scales with relative saliency. The highest performance is observed for PartImageNet, which contains only one salient object per image. Intermediate performance is observed for Pascal Part, which features multiple objects with varying saliency. The lowest performance is observed for PACO, which includes many objects with lower saliency. This ordering holds for all granularities and consistency measures. A plausible explanation is that less salient regions often occupy fewer pixels, so even small boundary misclassifications can have a larger effect on Intersection-over-Union scores. Interestingly, PACO is the only dataset in which object-level IoU is lower than part-level IoU. One possible explanation is that PACO's object labels can be more visually ambiguous, such as a table full of food, making it more difficult for the model to accurately delineate the full object boundary without including unwanted content. In contrast, certain parts might be

Method	Pascal Part [1]			PartImageNet [9]			PACO [30]		
	$\text{mIoU}_{\text{Obj}} \uparrow$	$\text{mIoU}_{\text{Part}} \uparrow$	CVR@0.5 $\downarrow$	$\text{mIoU}_{\text{Obj}} \uparrow$	$\text{mIoU}_{\text{Part}} \uparrow$	CVR@0.5 $\downarrow$	$\text{mIoU}_{\text{Obj}} \uparrow$	$\text{mIoU}_{\text{Part}} \uparrow$	CVR@0.5 $\downarrow$
conSAMmé-Vanilla	70.11	**36.17**	32.65	76.19	42.78	**17.06**	21.37	**36.63**	42.40
conSAMmé	**71.95**	36.11	**32.21**	**76.27**	**42.99**	17.43	**21.52**	35.92	**40.56**

Table 2. Comparison of conSAMmé with and without contrastive learning (conSAMmé-Vanilla). We observe that contrastive generates more performant object segmentations, while increasing consistency in complex scenes.

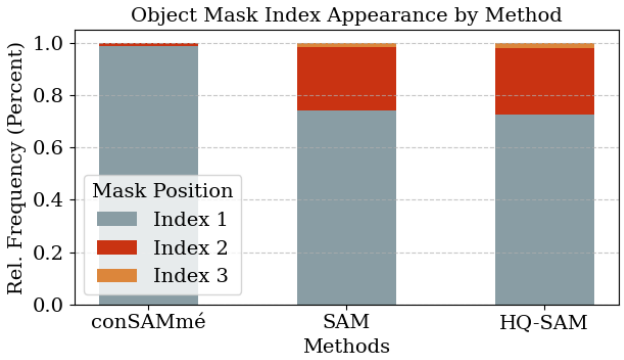

Figure 4. Stacked bar chart showing relative frequency of object mask placement across three output positions. conSAMmé consistently places the object mask in the first position, while SAM and HQ-SAM show greater variability, distributing the mask more evenly across positions.

more visually distinct (for instance, a smaller, more texture-specific or shape-specific region such as a table leg), making them easier to segment despite lower saliency. This difference could lead to instances where part-level segmentation outperforms object-level segmentation in terms of IoU.

Analysis with respect to segmentation consistency. Across all datasets, conSAMmé consistently outperforms the baselines in consistency, as measured by the CVR metric. On average, it produces similar object masks for each part when prompted with a click. For instance, on PartImageNet, conSAMmé nearly doubles the consistency performance compared to the next best baseline, HQ-SAM. This suggests conSAMmé's outputs are less sensitive to click location. We attribute this robustness to our episodic click sampling strategy, which trains the model to segment from varied click positions using only a single click. Unlike traditional methods that optimize for minimal clicks to reach a target IoU, our approach focuses on maximizing initial IoU, potentially reducing user effort during refinement.

Consistency, like segmentation quality, scales with dataset saliency. PartImageNet, composed of salient regions, shows the highest consistency, followed by Pascal Part and PACO. Lower saliency correlates with cluttered or complex scenes, which hinders object distinction and increases variability across clicks, reducing consistency.

Analysis with respect to mask ordering. To assess how consistently each model places object and part masks in their intended positions, we sort the outputs by size and use Hungarian Matching [16] to identify the best match to each ground truth object and part. We focus on PartImageNet, where overall segmentation performance is stronger than in PACO, so we can isolate the issue of mask ordering. We plot the relative frequency with which each method positions the object mask across its available output positions. Results are shown in **Figure 4**. Although many methods only produce two masks for part-object segmentation, SAM and its variants can output three, meaning the object mask *could* appear in the last position as well.

Among all methods, conSAMmé is the only one that nearly always locates the object mask in the first position (about 99% of the time). By contrast, SAM places the object mask first 74% of the time, second 24%, and occasionally last. HQ-SAM shows a similar pattern. This variability suggests that users of baseline methods may have to search among multiple masks to find the desired granularity. By contrast, conSAMmé not only boosts consistent object placement but also maintains a coherent hierarchy for parts, as reflected in its lower CVR scores. This consistency can reduce user overhead, since the same output positions can almost always be associated with specific hierarchy levels.

Qualitative Results. We show examples of how each method responds to clicks on different parts of an object in **Figure 5**.

These qualitative results reinforce the quantitative observation that conSAMmé yields more consistent and precise object masks, independent of click location. In the first two rows, clicking on the bird's head leads all methods to correctly segment the object, but SAM and HQ-SAM mistakenly include the floor when the click is on the tail. In contrast, conSAMmé produces consistent masks across both clicks, omitting irrelevant regions, and instead staying within ground truth boundaries when mistakes are made. A similar trend appears in the complex cat scene: clicks on the head cause SAM and HQ-SAM to include parts of the computer, whereas conSAMmé accurately segments the cat. All methods perform well when the click is on the cat's feet. For part segmentation, conSAMmé successfully isolates specific parts (e.g., head/tail of the bird, feet/head of the cat), while baseline methods include extraneous content.

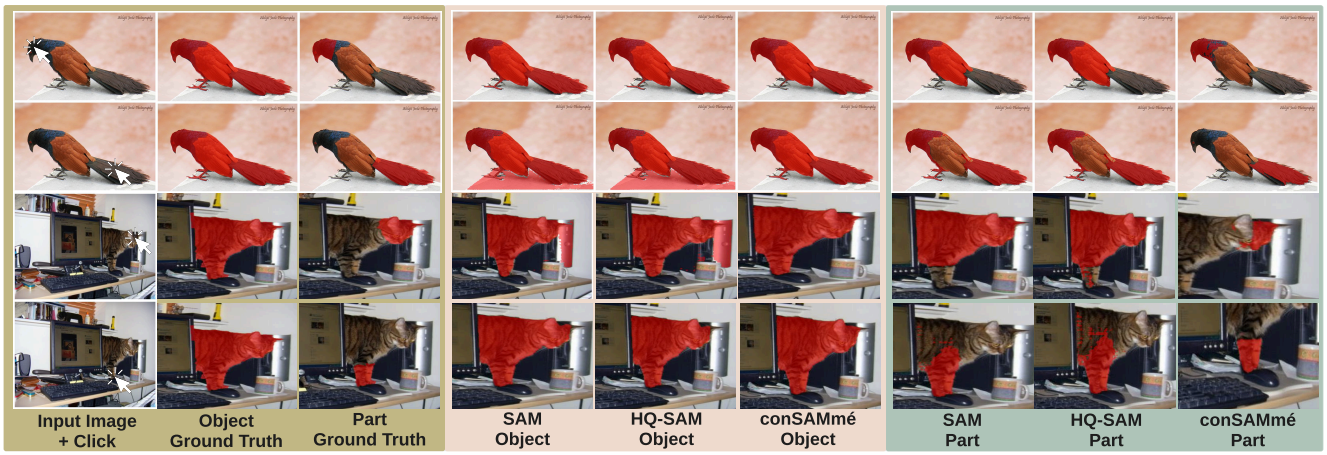

Figure 5. For both a single-object scene (a bird) and a complex scene, clicking on different parts yields varying masks with SAM and HQ-SAM: e.g., the floor is included when clicking the bird's tail but not its head. In contrast, conSAMmé produces consistent masks across clicks and achieves superior part segmentation. For the complex scene, segmented regions are cropped and scaled for clearer visualization.

Still, conSAMmé has some minor errors. For example, occasionally, it slightly misses parts of objects such as the bird's feet when the head is clicked, which we hypothesize could be a result of their small size. We hypothesize that performance could be further improved with more part annotations during training.

4.3. Model Design Analysis

To analyze the effectiveness of the contrastive learning portion of our proposed framework, we compare our approach against a baseline: conSAMmé without the addition of contrastive learning (i.e., only optimizing Equation 2). We call this conSAMmé-Vanilla. We provide results on all three datasets in **Table 2**. We note that both conSAMmé and conSAMmé-Vanilla outperform SAM with respect to both performance and consistency, highlighting the effectiveness of our framework components.

Effect of Contrastive Learning on Performance. When evaluating conSAMmé's segmentation performance, we observed mixed outcomes. In every setting, conSAMmé achieved superior object segmentation, with the most substantial improvements on Pascal Part, which features multiple salient regions. However, its part-level performance declined slightly. We hypothesize that hierarchical conditioning may bias the model toward object-level consistency, leading to less attention on finer part boundaries. This trade-off suggests that while hierarchical cues enhance robustness for larger entities, refining the quality of our weakly supervised part-level data is a valuable direction for future work to boost part segmentation, particularly since our contrastive loss relies on that supervision signal.

Effect of Contrastive Learning on Consistency. On Pascal Part and PACO, conSAMmé achieves more consistent outcomes, with larger gains on PACO. The contrastive learning framework improves robustness to variations in click positions on an object. On PartImageNet, which contains only one salient object, conSAMmé shows a slight decrease in object consistency. In simpler scenes, contrastive learning offers fewer advantages and introduces minor overhead that reduces overall consistency.

Despite the slight reduction on PartImageNet, conSAMmé delivers substantial gains on complex, multi-object datasets like Pascal Part and PACO. In these scenarios, multiple salient regions and varying object scales increase the likelihood of errors when methods rely solely on scale cues. By conditioning on hierarchical semantics, conSAMmé handles diverse object configurations more robustly, maintaining coherent segmentations when clicks differ across parts of the same object.

5. Conclusion

We present conSAMmé, a weakly supervised, contrastive framework that enhances consistency in multi-output interactive segmentation. Our method leverages an episodic click sampling strategy so that similar clicks, such as those on an object's part, produce a comparable output mask, such as the object itself. We also introduce a new evaluation metric for measuring model consistency and show that conditioning on hierarchical semantics yields more robust, coherent results than baseline methods.

Acknowledgments. Josh Myers-Dean is supported by a NSF GRFP fellowship (#1917573) and completed this work during an internship with Adobe Research.

References

[1] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 2, 5, 6, 7

[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. 1, 2, 3

[3] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 1

[4] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 2

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words. *arXiv preprint arXiv:2010.11929*, 7, 2020. 3, 4, 5

[6] Katy Ilonka Gero, Chelse Swoopes, Ziwei Gu, Jonathan K Kummerfeld, and Elena L Glassman. Supporting sensemaking of large language model outputs at scale. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2024. 2

[7] Han Gong and DP Cosker. Interactive shadow removal and ground truth for variable scene categories. In *BMVC 2014-Proceedings of the British Machine Vision Conference 2014*, 2014. 1

[8] Han Gong and Darren Cosker. User-assisted image shadow removal. *Image and Vision Computing*, 62:19–27, 2017. 1

[9] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022. 2, 6, 7

[10] Duojun Huang, Xinyu Xiong, Jie Ma, Jichang Li, Zequn Jie, Lin Ma, and Guanbin Li. Alignsam: Aligning segment anything model to open context via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3205–3215, 2024. 2

[11] Siyi Jiao, Wenzheng Zeng, Changxin Gao, and Nong Sang. Dfimat: Decoupled flexible interactive matting in multi-person scenarios. In *Proceedings of the Asian Conference on Computer Vision*, pages 2988–3004, 2024. 1

[12] 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. 2, 5, 6

[13] Eric Kee, Adam Pikielny, Kevin Blackburn-Matzen, and Marc Levoy. Removing reflections from raw photos. *arXiv preprint arXiv:2404.14414*, 2024. 1

[14] Scott Kelby. *Professional portrait retouching techniques for photographers using photoshop*. Pearson Education, 2011. 5

[15] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 2, 3, 6

[16] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 7

[17] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In *European Conference on Computer Vision*, pages 70–88. Springer, 2024. 2

[18] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27948–27959, 2024. 1

[19] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 577–585, 2018. 2

[20] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, Sim-Heng Ong, and Jiashi Feng. Multiseg: Semantically meaningful, scale-diverse segmentations from minimal user input. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 662–670, 2019. 1, 2, 5

[21] Kangning Liu, Brian L. Price, Jason Kuen, Yifei Fan, Zijun Wei, Luis Figueroa, Krzysztof J. Geras, and Carlos Fernandez-Granda. Uncertainty-aware fine-tuning of segmentation foundation models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 2, 5

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 1

[23] Xiaoning Liu, Zongwei Wu, Ao Li, Florin-Alexandru Vasluianu, Yulun Zhang, Shuhang Gu, Le Zhang, Ce Zhu, Radu Timofte, Zhi Jin, et al. Ntire 2024 challenge on low light image enhancement: Methods and results. *arXiv preprint arXiv:2404.14248*, 2024. 1

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[25] Sepideh Sarajian Maralan, Chris Careaga, and Yagiz Aksoy. Computational flash photography through intrinsics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16654–16662, 2023. 1

[26] S Mahdi H Miangoleh, Zoya Bylinskii, Eric Kee, Eli Shechtman, and Yağiz Aksoy. Realistic saliency guided image enhancement. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 186–194, 2023. 1

[27] Josh Myers-Dean, Yifei Fan, Brian Price, Wilson Chan, and Danna Gurari. Interactive segmentation for diverse gesture types without context. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7198–7208, 2024. 2

[28] Josh Myers-Dean, Jarek Reynolds, Brian Price, Yifei Fan, and Danna Gurari. Spin: Hierarchical segmentation with subpart granularity in natural images. In *European Conference on Computer Vision*, pages 275–292. Springer, 2024. 2

[29] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8743–8756, 2022. 5

[30] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023. 2, 6, 7

[31] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 5

[32] 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. 1, 3

[33] Shuyang Sun, Runjia Li, Philip Torr, Xiuye Gu, and Siyang Li. Clip as rnn: Segment countless visual concepts without training endeavor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13171–13182, 2024. 2

[34] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3635–3647, 2024. 2, 3

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34: 12077–12090, 2021. 1

[36] 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. 1, 3

[37] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023. 2

[38] Huimin Zeng, Jie Huang, Jiacheng Li, and Zhiwei Xiong. Region-aware portrait retouching with sparse interactive guidance. *IEEE Transactions on Multimedia*, 26:127–140, 2024. 1

[39] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 4

[40] Tianfei Zhou and Wenguan Wang. Cross-image pixel contrasting for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1