Appendix

In this Appendix, we summarize the notations of variables in Table 2 and describe the inference process in Algorithm 1. We then provide additional discussion of related concepts and methods, and present an ablation study to evaluate the uncertainty guided interactions. We also provide illustrative examples based on real user interactions.

Table 2: Summary of Notations

$\mathcal{T}_t$	task t
X_t	Images in task t
A_t, U_t	ground truth labels of annotated and unannotated pixels in task t
h_k^l	k-th channel of the feature map at layer l
W_k^l	k-th convolutional kernel at layer l
z_k^l	binary mask controlling layer l kernel k's activation
$a_{k,t}^l, b_{k,t}^l$	Beta parameters of v_k^l in task t
$\mu_{ki}^l, (\sigma_{ki}^l)^2$	Gaussian mean and variance of W_k^l's element i in task t
π_k^l	Bernoulli parameter of z_k^l
$\mathbf{r}_m, \hat{\mathbf{r}}_m$	RGB vector and predicted RGB vector of pixel m
$\bar{\boldsymbol{\theta}}_m$	mean predicted probability vector of pixel m
$\mathbf{y}_m$	ground truth one-hot label vector of pixel m
$\boldsymbol{\theta}_m^*$	propagated probability vector of pixel m
$\mathbf{y}_m^*$	propagated one-hot label vector of pixel m

Algorithm 1 Inference Process for Task t

Given new image X_t;
Given parameters $a_{k,t-1}^l$, $b_{k,t-1}^l$, $\pi_{k,t-1}^l$, $\mu_{ki,t-1}^l$, $\sigma_{ki,t-1}^l$;
Generate embedding of image e via variational autoencoder;
Retrieve the nearest neighbor e^* from recorded embeddings and use the corresponding mask z^*;
do
  for layer l=1:L **do**
    for kernel k=1:K **do**
      Sample $W_{k,t}^l$ using (3);
    end for
  end for
  Predict segmentation and uncertainty using (10),
  Collect user annotations and propagate using (11);
  for layer l=1:L **do**
    for kernel k=1:K **do**
      Sample $v_{k,t}^l$, $z_{k,t}^l$, $W_{k,t}^l$ using (2), (8), (3);
    end for
  end for
  Evaluate loss using (6), (15);
  Update $a_{k,t}^l$, $b_{k,t}^l$, $\mu_{ki,t}^l$, $\sigma_{ki,t}^l$, $\pi_{k,t}^l$;
while user unsatisfied with segmentation
Record image embedding e and mask z;

Discussion of Related Concepts

We further differentiate the proposed CLIS framework from some important related concepts, focusing on the problem setting, model training, and performance evaluation. We then highlight the key technical difference between CLIS and some specific methods as compared in the main paper.

Table 3 summarizes the difference between CLIS and related concepts, including conventional interactive segmentation, continual learning with new classes (CL-new class), continual learning with new instances (CL-new instance), active learning, and online learning.

- CL-new class: Continual learning with new classes, or incremental class learning, is one scenario of continual learning. It formulates a series of tasks in which some new classes are available to the model. The model should be able to recognize the new classes without forgetting the previous ones.
- CL-new instance: Continual learning with new instances is another scenario of continual learning. All classes are known in pre-training, and each subsequent task may contain instances with a different subset of classes.
- Active learning: An active learning algorithm allows human interaction by querying a user to label new data points to improve the algorithm's performance efficiently. Active learning can be applied to interactive segmentation, and it typically queries a user to provide pixel-wise labels of a whole image. It is different from interactive segmentation, where a user only annotates a few pixels.
- Online learning: An online learning algorithm aims to optimize the predictions over data instances sequentially. It should be noted that online learning also suffers from catastrophic forgetting problems, which is similar to continual learning. However, online learning does not emphasize task-based setups, and keeps learning over time in a streaming fashion.

Generally speaking, the setting of CLIS is similar to CL-new instance, but is customized for interactive segmentation, because it processes data in small batches and actively queries user annotation on selective pixels.

Ablation Study

We conduct an ablation study to analyze the effectiveness of uncertainty guided interaction. Given uncertainty estimation, we simulate user interactions to annotate the sixteen erroneous segmented areas with the highest uncertainty. An alternative strategy without considering uncertainty estimation is to randomly select erroneous segmented areas and provide the correct labels. We compare the two strategies in terms of refined segmentation results. The results in Figure 5 show that uncertainty-guided annotations consistently outperform the strategy of randomly choosing erroneous segmented areas and providing annotations.

Illustration of Real-User Interaction

We provide illustrative examples of the user interactions in Figure 6. Given new images, the model generates initial segmentation (column 3). Then the users annotate on selective areas (column 5) guided by uncertainty information (column 4). After that, the model refines segmentation results (column 6) based on user annotation. As shown in Figure 7, the user interface includes a pop-up window visualizing the segmentation map, which allows users to click for annotation,

Table 3: Comparison on problem setting, model training, and performance evaluation

	How tasks are formulated	How model interact with users	Require full/partial label for images
CLIS	New instances introduced per task	Actively query user annotation	Partial label
Interactive Segmentation	No specific task formulation	Actively query user annotation	Partial label
CL-new class	New classes introduced per task	N/A	Full label
CL-new instance	New instances introduced per task	N/A	Full label
Active Learning	No specific task formulation	Actively query user annotation	Full label
Online Learning	No specific task formulation	N/A	Full label

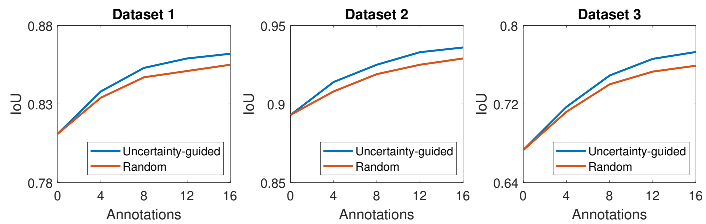

Figure 5: Ablation study that compares uncertainty guided interaction with random annotations on the erroneous segmented areas: segmentation IoU is recorded after different numbers of annotations.

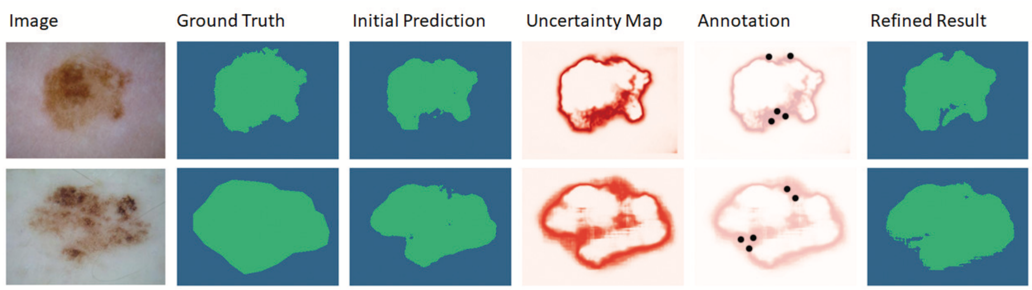

Figure 6: Illustrative examples of uncertainty-aware interactive segmentation: Different semantic classes are denoted by different colors in segmentation maps (columns 3 and 6). Areas with high uncertainty are highlighted in red (column 4), and annotated areas are highlighted in black (column 5).

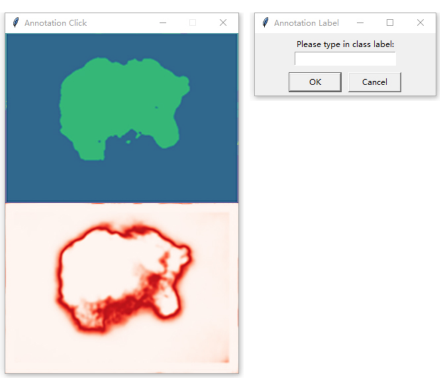

Figure 7: An illustrative example of user interface: users click on the pop-up window and provide the corresponding pixel label on the textbox.

and a textbox for typing in the corresponding labels. Then the information of annotations is passed back to the model for refinement.