Interact with Open Scenes : A Life-long Evolution Framework for Interactive Segmentation Models

Ruitong Gan
ganruitong2020@ia.ac.cn
School of Artificial Intelligence, University of Chinese Academy of Sciences
Center for Research on Intelligent Perception and Computing, CASIA
Beijing, China

Junsong Fan
Yuxi Wang
fanjunsong2016@ia.ac.cn
wangyuxi2016@ia.ac.cn
Center for Research on Intelligent Perception and Computing, CASIA
Beijing, China
Center for Artificial Intelligence and Robotics, HKISI_CAS
HongKong, China

Zhaoxiang Zhang*
zhaoxiang.zhang@ia.ac.cn
Center for Research on Intelligent Perception and Computing, CASIA
Beijing, China
Center for Artificial Intelligence and Robotics, HKISI_CAS
HongKong, China

ABSTRACT

Existing interactive segmentation methods mainly focus on optimizing user interacting strategies, as well as making better use of clicks provided by users. However, the intention of the interactive segmentation model is to obtain high-quality masks with limited user interactions, which are supposed to be applied to unlabeled new images. But most existing methods overlooked the generalization ability of their models when witnessing new target scenes. To overcome this problem, we propose a life-long evolution framework for interactive models in this paper, which provides a possible solution for dealing with dynamic target scenes with one single model. Given several target scenes and an initial model trained with labels on the limited closed dataset, our framework arranges sequentially evolution steps on each target set. Specifically, we propose an interactive-prototype module to generate and refine pseudo masks, and apply a feature alignment module in order to adapt the model to a new target scene and keep the performance on previous images at the same time. All evolution steps above do not require ground truth labels as supervision. We conduct thorough experiments on PASCAL VOC, Cityscapes, and COCO datasets, demonstrating the effectiveness of our framework in solving new target datasets and maintaining performance on previous scenes at the same time.

CCS CONCEPTS

• **Computing methodologies** → **Image segmentation**; *Scene understanding*.

KEYWORDS

Computer Vision, Interactive Segmentation

*Corresponding author

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10.
https://doi.org/10.1145/3503161.3548131

ACM Reference Format:
Ruitong Gan, Junsong Fan, Yuxi Wang, and Zhaoxiang Zhang. 2022. Interact with Open Scenes : A Life-long Evolution Framework for Interactive Segmentation Models. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3548131

1 INTRODUCTION

In the past few years, with the rapid progresses of deep learning techniques, enormous improvements have been made in computer vision fields including semantic segmentation [3, 16, 23, 27], instance segmentation [1, 10] and object detection [6, 19, 37], etc. Interactive segmentation is a sub-task of semantic segmentation, expecting the model to generate high-quality masks with the help of limited user interactions. Many researchers have concerned the efficiency of different interactive methods, such as iterative clicks [21, 25, 45], bounding boxes [35, 44], and other specified points [17, 26, 47]. Some other approaches pay their attention to the segmentation quality by improving interactive information encoding [22, 29], applying multi-scale features [18, 31], and inserting attention mechanisms [2, 9] to the models, etc. These methods are trained with precise ground truth masks and achieved impressive results with various testing policies.

However, there are a large number of images with different distributions and patterns in real-world application scenes, such as images from synthetic and reality, images taken from different camera params, etc. With the original settings for interactive segmentation tasks, if the model wants to segment a certain set of images, it should be trained on a corresponding domain with precise label masks. As shown in Fig. 1(a), previous interactive segmentation methods trained their models with ground truth labels and tested with user interactions on the same source dataset. But obtaining such a large amount of ground truth labels is impractical due to costly annotation and economic burden. Another obvious strategy to solve such problem is the Domain Adaptation (DA) method. As shown in Fig. 1(b), the model is trained on a labeled source dataset and tested on unlabeled target images. However, a model is only available for segmenting pre-defined specific datasets at each moment, while is unable to deal with dynamic scenes encountered.

Considering all factors mentioned above, a continual adaptation strategy can be applied to interactive segmentation tasks in order to

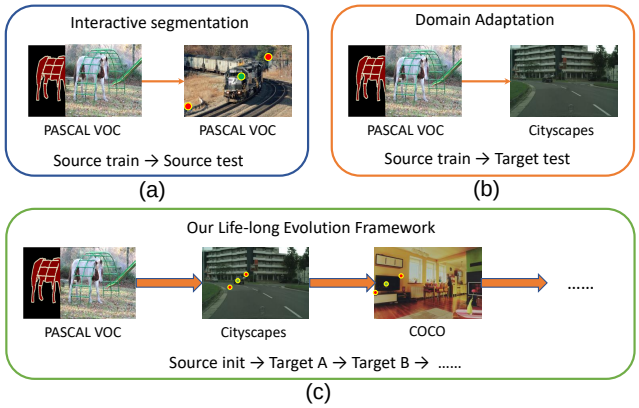

Figure 1: Comparison of dataset management between (a) previous interactive segmentation, (b) domain adaptation, and (c) our life-long evolution framework.

allow the models to gradually improve their performance on target scenes with various new styles and classes. Besides, adopting user interactions is a natural and easy way to provide weak supervision for further evolution of the model instead of requiring the costly annotation burden and additional user efforts. Thus we establish a life-long evolution framework for the interactive segmentation model, making it possible to become a universal model and handle images with different patterns or distributions with itself, the concept is shown in Fig. 1(c). We use limited initial data with ground truth to train an initial interactive model. The initial model is used to generate coarse pseudo masks on the target dataset, which are refined with our Interactive-ProtoType module. After that, the model is further evolved with images from both the target set and initial data with alignments applied among features. When our framework is applied to real application situations, it can obtain abundant free user interactions among the usage of our model, with which the model can be further off-line evolved. The new evolved model can provide better results on images that users applied while still keep the performance on former datasets.

We conduct extensive experiments with the Pascal VOC 2012 [8], Cityscapes [5] and MS COCO [20]. We train the initial model on PASCAL VOC dataset and test on the Cityscapes and COCO datasets. Our approach could boost the model's mIOU performance from 73.7% to 79.4% and from 74.9% to 80.1% respectively, which is a new state-of-the-art result in this field. In summary, the main contributions of the paper are:

- We are the first to propose a life-long evolution framework for interactive segmentation models, which takes freely available user interactions as supervision to guide the evolution without any further pixel-level ground truth masks. The evolved model has the ability to segment different target scenes with distribution variety, and can maintain the performance on previous witnessed images.
- An interactive-prototype module and a feature alignment module are designed as the core components to extract reliable supervision from weak interaction hints and utilize the information from previous images.
- Our experiments indicate that our framework can boost the performance of the model on Cityscapes to 79.4% mIOU and on COCO to 80.1% mIOU, which is a new state-of-the-art result.

2 RELATED WORKS

2.1 Interactive Segmentation

Interactive segmentation has a different type of network input compared with semi-supervised [49–51] and unsupervised segmentation [28, 41, 43] , as it concatenates user interactions as additional information while training and testing. The types of interactions provided vary from each other, such as clicks [21, 25, 45], bounding boxes [35, 44], points [17, 26, 47], etc. Jiang et al.[13] provides a back-propagating refinement scheme to correct mislabeled pixels based on multiple-round user interactions. Lin et al. [21] emphasizes the importance of the first click in segmentation networks during the entire user interaction period, where they introduced an additional first click loss to supervise original click loss. These methods reach comparably better results while requiring multiple times of user inputs and are time-consuming. Maninis et al. [26] chooses four extreme points from each location of the object-of-interest, obtaining bounding boxes from interactive points. However, when coming up with long, thin, leaning objects, clicks will be redundant and time-consuming for users. Zhang et al. [47] uses two diagonal points for box selection and a foreground point to mark the object. These methods have a unique pair of clicks for every single object, but they fail to utilize the annotations to optimize the model in open scene datasets.

Apart from traditional interactive segmentation methods, there are also recent works that adopt continual learning strategies as their methods. Zheng et al. [48] formulates a continual learning problem for interactively improving the segmentation results on new images in a known-classes dataset. However, the key structure of this paper is indeed a semantic segmentation framework which takes user annotations as weak labels to refine the segmentation results. Theodora et al. [15] points out that user corrections can be used as training examples to update the model, but they require a specific finetuned model on each dataset, and the class information from the ground-truth labels is restricted for alignments in the experiments. These methods mainly focus the continual learning sequences on a limited specific dataset pair, but fail to solve open target scenes with different classes and patterns.

2.2 Domain Adaptation

Domain adaptation (DA) tasks focus on transferring models from source data to target data[39, 46]. The images in source dataset and target dataset have the same classes, while hold different domain distributions. A typical dataset setting for Domain Adaptation tasks is adapting a model trained on synthetic images [33] to target real-world datasets [5]. To solve the problem, some methods [14, 24, 32] use Maximum Mean Discrepancy to measure the distribution divergence between different domains. Other researchers adopt conditional distribution alignments among classes [14, 34], separating classes into stuff and thing and making alignments accordingly [42]. In DA tasks, each model is trained only suitable for

a single pair of the source-target domain, which lacks the ability to adapt multiple target sets at the same time.

2.3 Pseudo-Mask Based Method

Pseudo masks are now widely used in Unsupervised training [30, 38, 40]. With a large amount of unlabeled data, pseudo masks are generated by trained models to enlarge the labeled training set, which is also known as Self-Training. When discussing an open scene dataset, pseudo masks can quickly help ease the domain shifting problems. Currently, most Domain Adaptation problems need target domain pseudo masks to help generate much better results, thus the quality of pseudo masks is rather important. Wang et al. [42] generates pseudo masks based on confidence score and calculates global adaptation loss to bridge the domain gap. Snell et al. [36] points out that pseudo masks need to be refined to prevent noisy labels from influencing results. They calculated the prototype of each class and weigh the distance of each class pixel to refine temporal pseudo masks.

3 METHODS

This section contains four parts. In Sec. 3.1, we introduce our life-long evolution framework given an initialized model trained on a limited amount of images with ground truth labels. In Sec. 3.2, we describe a pseudo masks refinement process called the *Interactive-ProtoType module* (IPT module), where the refined masks are used as self-training labels in further steps. Sec. 3.3 introduces the fine-tuning process after obtaining the refined pseudo masks, and the feature alignment module applied within training. In the final Sec. 3.4, we emphasize some details about our evolution framework when dealing with multiple datasets.

3.1 Framework Architecture

The overview of our framework is shown in Fig. 2. At the start of our framework, we only have a limited number of images with ground truth labels, which is considered as the initial dataset $\mathcal{A}$. We follow the interactive strategy in IOG [47], two diagonal background points shaping the bounding box and one foreground point for each object. After receiving the interactive point pair from the user, two Gaussian maps are generated based on the foreground and background points respectively. The two Gaussian maps are then concatenated with the RGB image into a 5-channel input, which trains the model in the initializing stage. This model is considered as the initial model trained on $\mathcal{A}$ with ground truth.

While the initial model processes new images in practical environments, our framework collects the new images and corresponding user interactions. They are then applied for life-long evolution without any ground truth masks so that the model can better fit practical environments. The collection of new images is denoted here as dataset $\mathcal{B}$. The proposed IPT module is responsible for generating refined pseudo-masks on $\mathcal{B}$, as illustrated in the blue part of Fig. 2. The initial model first inferences coarse pseudo masks for each object based on interactions. The network output feature and coarse mask are then calculated into three prototypes using interactions as hints. Prototypes are used to filter the confidence scores of each pixel in the feature and turn into confidence label maps of the same size as the coarse pseudo mask. A voting method is applied to select each pixel in the final refined mask. The IPT module is responsible for producing pseudo masks for new data and does not update the model parameters.

After obtaining the refined pseudo masks on $\mathcal{B}$, we then finetune the initial model in the Feature Alignment module. Both images in $\mathcal{A}$ and $\mathcal{B}$ are sent into the network to extract the features. Images from $\mathcal{A}$ are supervised with ground truth labels while the images in $\mathcal{B}$ use refined masks as supervision, calculating a Maximum Squares Loss [4] and a Feature Alignment loss.

3.2 Interactive-ProtoType Module

We first acquire the model from the initializing stage, and we have the RGB images as dataset $\mathcal{B}$ with user interactions on each object. Previous work [26, 47] only concatenates Gaussian maps generated from interactions as input in order to make use of the interactions. However, we argue that these interactive point pairs have abundant information **beyond** simply generating the Gaussian maps as supervision. Based on the network structure in IOG [47], we add a pixel prediction layer after the fourth convolution block of the Resnet [11] structure, and in the IPT module this predictor is a binary mask predictor, which takes image features from the segmentation backbone as input, and outputs a 0-1 value binary mask. The output of the predictor is a 2D mask representing the foreground possibility of each pixel. Thus we can obtain three components from the network output with the same spatial size $H \times W$: a coarse pseudo mask $\mathcal{M}_p$ from the segmentation prediction, a binary predictor output $\mathcal{M}_b$ and an image feature F from the fourth layer of the Resnet model.

To utilize the information from the above components, we calculated three different prototypes with additional information from the foreground interactive point: global average prototype $c^{(g)}$, center point prototype $c^{(c)}$ and edge prototype $c^{(e)}$. Prototypes can be calculated with :

$$c = \frac{1}{Z} \sum_{i=1}^{HW} Q_i F_i, \tag{1}$$

where F_i is the vector in feature map F on pixel i, Z is the value sum of Q. For $c^{(g)}$, $Q = \mathcal{M}_p \odot \mathcal{M}_b$ is the hadamard product of $\mathcal{M}_p$ and $\mathcal{M}_b$, so $c^{(g)}$ is the mean vector value in image feature with all the pixels considered as foreground. For $c^{(c)}$, we randomly sample x foreground points around the original foreground point provided by the user inside the coarse pseudo mask, and consider the mean value of these $x+1$ point vectors as $c^{(c)}$, so here Q is a binary map where positions of the $x+1$ points are set to 1. For $c^{(e)}$, the coarse pseudo mask $\mathcal{M}_p$ is eroded in order to extract the edge mask $\mathcal{M}_e$ representing the edge position of $\mathcal{M}_p$. Here Q in Eq. 1 is $\mathcal{M}_e$.

The squared Euclidean distance between a prototype and a vector of pixel i in feature map F is calculated as:

$$d(c, F_i) = \|c - F_i\|^2, \tag{2}$$

so the confidence map of a prediction from a prototype is generated as follows:

$$P_i = \frac{\exp(-d(c, F_i))}{\exp(-d(c, F_i)) + \exp(-d(\bar{c}, F_i))}, \tag{3}$$

where P is the predicted confidence map shaped as $1 \times H \times W$, $\bar{c}$ is calculated with Eq.(1) by reversing Q to $\bar{Q} = 1 - Q$.

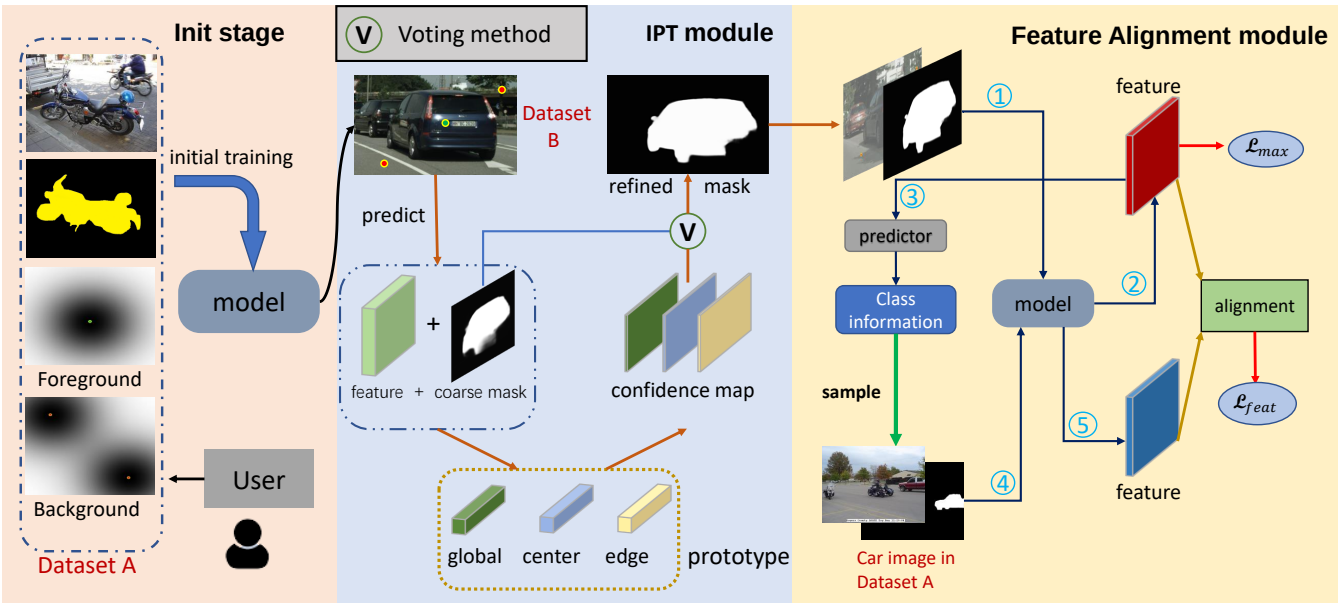

Figure 2: An overview of our framework. Here, dataset A in the pink square has pixel-wise label masks, while dataset B in the blue square only contains weak user interactions. In the initializing (init) stage, the model is initialized on dataset A with ground-truth and interactions. The IPT module uses the trained model to generate coarse pseudo masks on dataset B, and follows the pipeline to refine the masks. In the feature alignment module, the model extracts features from images in both datasets A and B to arrange a feature alignment loss, and the network calculates the maximum square loss for the image in dataset B.

With the three prototypes, we have $P^{(g)}$, $P^{(c)}$ and $P^{(e)}$ indicating confidence maps respectively. Pixel in a confidence map is more likely to be considered as foreground point if its value is closer to 1. Mean confidence score k of each confidence map can be calculated with Eq. 4:

$$k = \frac{\sum_i P_i \mathcal{M}_{pi}}{\sum_i \mathcal{M}_{pi}}, \tag{4}$$

where i is the non-zero pixel index in pseudo mask $\mathcal{M}_p$. As the network already learned information from the center point Gaussian map, we then let $P^{(m)} = \alpha \times P^{(c)} + \beta \times P^{(e)}$ to merge the two confidence maps, with a higher merging weight for edge confidence map to emphasize more information in the edge pixels and remove possible stria around large objects. Now k_c, k_m is calculated from Eq. 4 respectively, and are further multiplied with parameters $m, n \in (0, 1)$. Pixel values in a confidence map bigger than the threshold will be judged as foreground pixels, and thus we obtain two final label maps $\mathcal{M}_c$ and $\mathcal{M}_m$ calculated from the global average prototype and edge & center merged prototype.

A voting mechanism is then applied to the existing three label maps $\mathcal{M}_p$, $\mathcal{M}_c$ and $\mathcal{M}_m$. Define $p = \mathcal{M}_p + \mathcal{M}_c + \mathcal{M}_m$, the final refined mask $\mathcal{M}$ is calculated with Eq. 5:

$$\mathcal{M}_i = \delta_i \times p_i, \tag{5}$$

where

$$\delta_i = \begin{cases} \frac{1}{p_i}, & p_i \geq 2 \\ 0, & otherwise \end{cases} \tag{6}$$

p_i is the pixel value in p. The refined mask $\mathcal{M}$ has the same shape as the coarse pseudo mask and is refined under the supervision of global, point, and edge information.

3.3 Feature Alignment & Training

Now that we have refined the pseudo masks on dataset $\mathcal{B}$ through the IPT module, in this section we explain the following training process and feature alignments among training.

In the Feature Alignment module the predictor consists of a segmentation head same structure as the head in the IPT module and an additional classifier head. The feature of the image from dataset $\mathcal{B}$ is extracted from the network backbone and is firstly sent into the classifier which has $n+1$ channels, and the output is a 1D class probability representing which class the object belongs to. Here n is the number of classes in dataset $\mathcal{A}$, and parameters for the first n channels of the classifier come from the previously trained n-channel classifier in the initial stage. The extra one channel is randomly initialized for the unseen classes in new data $\mathcal{B}$. This classifier is supervised in the alignment training process with the sampled images from dataset A.

When the predictor judges the image in $\mathcal{B}$ to be classes seen in $\mathcal{A}$, the model samples an image of the same class in $\mathcal{A}$ for further alignment. If the image is considered an unseen class, a random image is sampled to format a class-agnostic binary objectness alignment. Two image features from both datasets $\mathcal{A}$ and $\mathcal{B}$ are then segmented by the segmentation head and receive binary 2D masks as output for for calculating the segmentation loss.

After image features F_a, F_b, which represent features of image from $\mathcal{A}$ and $\mathcal{B}$ respectively, are extracted from network, , two global average prototypes c_a, c_b for each image can be calculated with Eq. 1, and a predictor output $\mathcal{M}_b^{'}$ shaped as $1 \times H \times W$ which is not binarylized, representing the possibility of the foreground point on each pixel. We arrange a feature alignment loss based on two prototypes from image pair of the same class, the loss function is shown in Eq. 7:

$$\mathcal{L}_{feat} = \|c_a - c_b\|^2 + \sum_i \frac{1}{\mathcal{M}_{bi}^{'}} \|F_{bi} - c_a\|^2, \tag{7}$$

where i is the non-zero pixel index in the pseudo mask $\mathcal{M}_p$, $\mathcal{M}_{bi}^{'}$ and F_{bi} is the value on the corresponding pixel position. When considering unknown class images, $\mathcal{L}_{feat}$ is simply calculated as:

$$\mathcal{L}_{feat} = \|c_a - c_b\|^2, \tag{8}$$

we remove the pixel-level alignment in the second part of Eq. 7, as the foreground pixels have different features compared with the randomly sampled images from dataset A. But we keep the squared Euclidean distance part of the loss function to align possible distribution similarity of the foreground object from both datasets and keep the model witnessing previous dataset images.

Additionally, we apply the Maximum Square Loss [4] to balance the gradient change. The loss function for known class images is defined as:

$$\mathcal{L}_{max} = -\frac{1}{2} \sum_{i=1}^{HW} \sum_{c=1}^{C} (p^{i,c})^2, \tag{9}$$

$p^{i,c}$ is the prediction value of class c at pixel index i in the predictor output. For the binary prediction layer of unknown class images, maximum square loss is calculated as:

$$\mathcal{L}_{max} = \sum_i (-p_i^2 - (1 - p_i)^2), \tag{10}$$

where i is the pixel index in the predictor output $\mathcal{M}_b^{'}$, p_i is the foreground possibility of pixel i.

With two binary cross entropy losses $\mathcal{L}_{ce}^a$, $\mathcal{L}_{ce}^b$ calculated from the image predictions, the final loss value for an image pair is shown in Eq. 11:

$$\mathcal{L} = \mathcal{L}_{ce}^a + \mathcal{L}_{ce}^b + \mathcal{L}_{feat} + \mathcal{L}_{max}. \tag{11}$$

3.4 Evolution Sequence on Multi-Datasets

In the previous sections, we describe the details of how our life-long evolution framework operates on two datasets. Here we will discuss how to arrange an evolution sequence on multiple datasets, which meets the real situation our framework is about to encounter. A brief concept figure of the evolution procedure is shown in Fig. 3.

With all of our modules mentioned in previous subsections, the first evolution step from initial dataset $\mathcal{A}$ to dataset $\mathcal{B}$ was completed successfully. The only difference in the following evolution steps is that all the previous images learned by the current model are merged into one combined dataset, and is randomly sampled in Sec. 3.3 when training with images from the new dataset. All images the model trained with only have interactive hints and refined pseudo masks except for the initial closed dataset, which has limited images and ground truth labels. If more images with ground truth labels are seen by our framework, they can be added to the merged dataset pool, enlarging the known classes and enhancing the segmentation results.

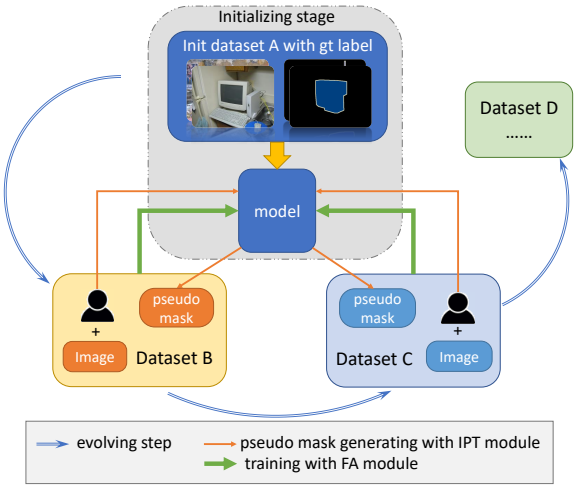

Figure 3: A brief concept figure of the evolution sequence on our framework with multiple datasets. The model is initialized with images and ground truth from dataset A in initializing stage, and evolves to datasets B and C gradually by generating refined pseudo masks and feature alignment training.

4 EXPERIMENTS

We conduct several evolution sequences for the initial interactive model trained on a closed dataset with ground truth labels. We keep settings from the previous work [47]: two diagonal background points and a foreground point representing the object of interest. The model of our framework is initialized on PASCAL VOC [8], and evolves to Cityscapes [5] and COCO [20] datasets in different settings. In Sec. 4.1 we introduce our datasets details, the parameter settings in our interactive-prototype module and the training processes. Sec. 4.2 is where we compare the performance and the evolution ability of our framework with some of the other interactive segmentation methods. Sec. 4.3 presents the ablation studies on each part of our framework. Note that the ground truth masks are never used in all our training procedures except initializing stage. The following experiments show that our method achieves promising results without the supervision of pixel-level masks.

4.1 Implementation Details

4.1.1 ***Datasets.*** We follow the same interactive strategy as previous work IOG [47]. All of our experiments in Sec 4.2 are using PASCAL VOC [8] augmented with SBD dataset as the closed training set for generating the initial interactive segmentation model, if without any further illustration. Cityscapes [5] dataset is considered to be the second interactive dataset, which contains 2,975 images with 52,004 unique objects. We do not follow the setting of COCO mini-val (COCO Mval) dataset used in previous works

[26, 47], as COCO Mval only contains 800 images with 800 instances, the amount of the images is too small to be involved in our evolution framework. Instead, we randomly sample 10% of the entire COCO2017 training set, including 11,901 images with 83,070 objects, while keeping the entire validation set as usual. All the COCO dataset mentioned in the following sections represents the sampled COCO2017 dataset, without exceptions.

4.1.2 ***IPT module.*** In the interactive-prototype module, we generate and refine the pseudo masks on target training sets from the network predictions of the initial model. The radius of Gaussian maps generated from the interaction points is set to 10. The parameter x mentioned in Sec. 3.2 when conducting the center point prototype is set to 3 in the IPT module. The merging weight for the edge confidence map is 0.7, higher than the 0.3 set for the center map. The erosion applied to the coarse pseudo masks is implemented by OpenCV package with the kernel size = 5×5 and iteration = 3. Through the experiments, the parameter values mentioned in Section 3.2, which are used to multiply k into threshold, show the best results for pseudo masks at $m = 0.70$, $n = 0.75$ respectively. The effectiveness of the IPT module will be shown in Section 4.3.1.

4.1.3 ***Training Details.*** Our initial model is trained on either PASCAL VOC only or PASCAL VOC augmented with SBD dataset for a maximum of 100 epochs. For further evolution steps, the model is trained on Cityscapes with 20 epochs, and on COCO with 10 epochs. All the processes in the IPT module are offline to prevent distribution fluctuation when seeing new domain features at the start. A loss percentage decay is also applied to the cross-entropy loss of the previous dataset, linearly decreasing to 0.1 throughout the training epochs, in order to ease the possible domain gap between different datasets. We use SGD as our optimizer, the learning rate, weight decay, and momentum are set to 1×10^{-9}, 5×10^{-4}, and 0.9, respectively. Among all the training steps, we set the batch size at 16, each image is randomly zoomed and rotated, cropped from the original image based on interactive points, and resized to 512×512. We use ResNet-101 [11] pretrained on ImageNet [7] as our image feature extractor backbone, and a additional pixel prediction layer is added after the output feature map of Resnet [11]. The number of classes set to the predictor follows the settings mentioned in Sec. 3.2 and Sec. 3.4 respectively. In our experiment PASCAL VOC has 21 classes, so the channel of predictor output is set to 22, adding an additional unknown class. When reprocessing the model following IOG [47] settings, we could only obtain model performance on Cityscapes at 77.2% mIoU, which is announced in the paper as 77.9% instead. All the experiments below are based on our reprocessed model, and evaluation results will be shown accordingly.

4.2 Compared with Previous Works

We first compare previous works on three benchmarks, *i.e.*, PASCAL VOC [8], Cityscapes [5] and COCO [20]. Table 1 shows the number of clicks required from different methods (or restricted by methods such as [26] and [47]) to reach a certain performance on each dataset, and a click mIoU is also measured for each method. The click mIOU is the value of the prediction results when restricting a method with a certain number of input clicks. Here we set the click number to be 3, as our method only simulates three points from the user, except DEXTR [26] which restricts their method to acquire four points. Both our results on Cityscapes and COCO come from the initial model trained on VOC and our evolution framework once, that is, VOC to Cityscapes and VOC to COCO. All the results listed in Table 1 are not finetuned by ground truth labels.

Table 1: Comparison with previous works. Here we set 3 as the restricted click number, and cIoU is the performance of each method with 3 interactive clicks.

Test Dataset	VOC@85%		Cityscapes@80%		COCO@85%	
Method	NoC	cIoU	NoC	cIoU	NoC	cIoU
iFCN[45]	6.9	-	-	-	8.06	-
Li et al.[17]	-	-	-	-	7.86	-
ITIS[25]	3.4	83.2	-	65.2	-	68.2
FCTSFN[12]	4.6	79.7	-	59.8	9.62	37.5
DEXTR[26]	4	91.5	4	76.4	4	78.3
IOG[47]	3	93.2	3	77.9	3	74.9
Ours	3	**93.2**	3	**79.5**	3	**80.1**

It can be seen from Table 1 that our framework achieves the improvement of 2.3% and 1.8% on Cityscapes and COCO compared with previous state-of-the-art respectively. This demonstrates the effectiveness of our evolution framework in dealing with datasets that differ from the initially trained dataset, and is able to achieve better performance without any pixel-wise masks finetuning.

To further examine the ability of our life-long evolution framework, we arrange several evolution processes using PASCAL VOC, Cityscapes, and COCO datasets. First, the model is trained on augmented PASCAL VOC and applied with our framework to Cityscapes, and then evolves to COCO eventually, simulating the situation when the model encounters different datasets in order, and testing its performance on all the three datasets after the evolution. We compare with IOG [47] and DEXTR [26], the experiment results are shown in Table 2. All the finetune steps of these two methods in Table 2 are using ground truth from the corresponding dataset, while our method keeps interactive information as hints and avoids ground truth supervision.

From the first column among Cityscapes and COCO dataset, it is shown that our life-long evolution framework can provide better results on Cityscapes and COCO dataset compared with previous works without any supervision of ground truth labels. And under each time step with *, which means that other previous works already finetuned to additional datasets with ground truth labels, our framework also maintains a promising score on VOC and Cityscapes after evolving more datasets respectively. The results of our method in the last column from Table 1 and Table 2 differ, as the model in Table 1 directly evolved from VOC to COCO, while another model runs an evolution sequence from VOC to Cityscapes, and COCO eventually.

4.3 Ablation Study

In the following subsections of the ablation study, most of the model, if not specified, is trained with ground truth only on PASCAL VOC

Table 2: The performance of our framework on three datasets compared with previous works. The Time step in the table denotes different training processes of the model. $t = 0$ is the initial model, $t = 1$ is the initial model finetuned to Cityscapes, $t = 2$ is $t = 1$ model finetuned to COCO. * denotes IOG and DEXTR are finetuned with ground truth labels, but ours is not.

Evaluated set		VOC			Cityscapes		COCO
Time Step		$t = 0$	$t = 1^*$	$t = 2^*$	$t = 1$	$t = 2^*$	$t = 2$
Methods	DEXTR[26]	91.5	87.8	88.4	76.4	76.9	78.3
	IOG[47]	93.2	87.3	87.6	77.9	78.4	74.9
	Ours	**93.2**	**92.6**	**92.7**	**79.5**	**79.3**	**79.8**

dataset as the initial model, and applied to Cityscapes or COCO with our life-long evolution framework step by step. We only use simulated user interactions instead of ground truth labels through our entire framework.

4.3.1 ***IPT module.*** After acquiring the initial model, the first step is to generate and refine the pseudo masks on Cityscapes training set. However, in order to follow the rule of not witnessing any ground truth labels of the images used in training, we test our method on the validation set of Cityscapes. We perform an ablation study on the Interactive-ProtoType module (IPT module) to validate the improvement of each step. The results are shown in Table 3. The global average prototype (GAP), center point, and edge maps bring an improvement on pseudo masks mIOU of 0.8%, 0.4%, and 0.6%, respectively, and have an entire performance boost of 1.1%. We then merge the edge and center point confidence maps with the weight of 0.7 and 0.3 respectively, along with the initial mask and the GAP confidence map to vote for the final pseudo masks in order to filter possible noises, and we receive an additional rise of 0.2% on mIOU. Methods without voting calculate final pseudo masks through weighted sum, with the weight of 0.8 for coarse pseudo masks and 0.2 for the rest of the maps equally. Fig. 4 shows the comparison of pseudo masks before and after our IPT module on Cityscapes and COCO dataset. From (c) in Fig. 4 it is clear that the eroded edge still has the mislabeled area on the right-top of the main object, but after taking the GAP confidence map into consideration and voting strategy, the refined mask correct that mistake. Though there are still hard cases such as mislabeled legs from the background person, the IPT module can indeed improve the quality of the pseudo masks.

4.3.2 ***Feature Alignment module.*** In this section, we will make ablation studies on the training procedure after obtaining the refined pseudo masks on target training sets, discussing the importance of each module and the different amount of data used for training the initial model. All the results announced in the following tables are based on Cityscapes and COCO validation set.

First, the experimental results shown in Table 4 are based on one single evolution process of our framework from PASCAL VOC to Cityscapes and COCO respectively. The initial model evaluates 73.7% and 74.9% mIOU without any further steps, which is the Baseline setting in the Table. After 20 epochs of self-training on Cityscapes and 10 epochs on COCO separately, using pseudo masks

Table 3: Ablation study on interactive-prototype module. The pseudo masks mIOU is evaluated on Cityscapes validation set.

Global	Center	Edge	Mer.& Vot.	mIOU
				73.7
√				74.5
	√			74.1
		√		74.3
√	√	√		**74.8**
√	√	√	√	**75.0**

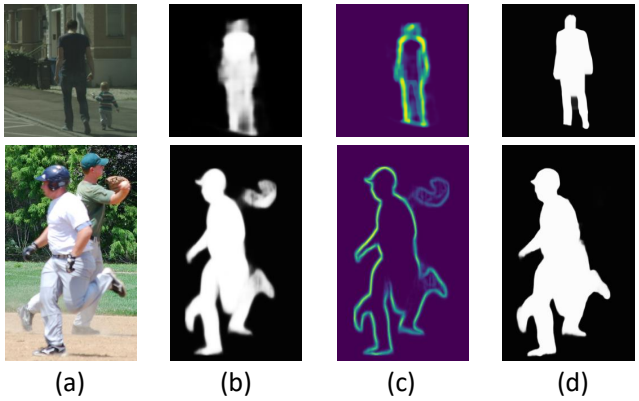

Figure 4: Visualization of images and pseudo masks refinement on Cityscapes and COCO. (a) is the original RGB image. (b) is the initial pseudo masks generated from the network. (c) is the heatmap of the eroded edge highlighting the edge prototype calculated area. (d) is the final refined pseudo mask.

generated from the initial model directly, its performance reaches 77.3% and 78.1% mIOU. For Cityscapes, the result has a 1.1% and 1.4% rise after adding the Interactive-ProtoType module (IPT) and feature alignment module (FA) respectively. While for COCO, the improvement for the result is 1.1% and 0.6%. The IPT module brings a better rise on COCO, while the FA module reaches a higher result on Cityscapes compared with each other. The reason for this situation is that all the classes in Cityscapes are already included in PASCAL VOC, which means that each object in Cityscapes images can be aligned with an object of the same class in PASCAL VOC and its ground truth. However, images in COCO have 80 classes compared with 21 classes in PASCAL VOC, and objects in COCO are more ambiguous and complex, thus the IPT module can contribute more to refining the stria among object edges. And when both IPT and FA module is applied, our model reaches the result of 79.4% and 80.1% in total. The last row of the table indicates the result of the initial model finetuned with ground truth, which is considered as upper bound of the task.

In order to figure out the effectiveness of our feature alignment loss function, we arrange an ablation study on Eq. 7, 8 to figure out how the loss function influences the results when witnessing unseen class images in a new dataset. We define $A = \|c_a - c_b\|^2$,

Table 4: Ablation study on training procedure and the upperbound of the model. Here ST means self-training, IPT denotes interactive-prototype module, FA means feature alignment module. Upperbound is the result of initial model finetune with ground truth. All the results are evaluated on Cityscapes validation set.

Model Settings (mIoU)	Cityscapes	COCO
Baseline	73.7	74.9
+ST	77.3	78.1
+ST +IPT	78.4	79.2
+ST +FA	78.8	78.7
+ST +IPT +FA	**79.4**	**80.1**
Upperbound	**82.2**	**82.6**

Table 5: Ablation study on feature alignment loss function. Three settings are applied to emphasize the necessity of different structures between seen and unseen class images.

	A	B	COCO	VOC
Component in Eq. 8	√	√	78.2	92.4
			79.7	91.9
	√		79.8	92.7

$B = \sum_i \frac{1}{M'_{bi}} \|F_{bi} - c_a\|^2$ in Table 4, thus the original Eq. 7 = A + B, Eq. 8 = A. We train the model on VOC datasets with ground-truth labels, and then evolve it to COCO dataset, the results are shown in Table 5. In the first row, the mIOU on COCO decreases because the pixel-wise alignment in the second part of Eq. 7 miss-match the features of two images of different classes, while the mIOU on VOC remains a comparable level as the image is sampled as usual. In the second row, the result on COCO has a minor difference, while mIOU on VOC drops as the knowledge from COCO cannot be utilized to improve the performance on VOC images. The last row is our original setting for Eq. 8.

We then establish a brief comparison of the different amounts of data usage with ground truth when training the initial model. The training epochs in the initializing stage are both set to 100. As shown in Table 6, when simply running the baseline without any other steps, the performances on both Cityscapes and COCO highlight the importance of the image amounts with the ground truth label used in the initial model training. Self-training strategy brings bigger improvement on models only initialized on VOC dataset, as these models learn fewer images at the previous step, which can be eased by the pseudo label supervision finetuning. After applying the entire methods in our framework, the original performance gap has been reduced to a comparable situation, indicating that our framework can overcome the lack of image & ground truth data usage when initializing the model.

In order to figure out how the number of new dataset images influences the learning process, we further divide the Cityscapes dataset into four random-selected subsets and train each one of them step by step to simulate a small evolution sequence within one dataset. We use the model initialized only on PASCAL VOC, and each subset is trained over 20 epochs from the previous evolved model, so after all four subsets are trained, the model learns the same amount of images compared with our settings previously mentioned in Sec. 4.1.3. As shown in Table 7, our model can improve its performance on the validation set from 73.7% to 78.3% with the help of only 25% of the entire images with user interactions. Through the sequence of additional images applied, the performance of our model further increases to the maximum of 79.3% mIOU, mainly reaching the result previously mentioned. Some minor fluctuation may have occurred due to the learning rate change at the beginning of each training procedure.

Table 6: Experiment on observing the influence of the amount of image & ground-truth usage in initializing stage.

eval dataset	Cityscapes		COCO	
init dataset	VOC	VOC + SBD	VOC	VOC + SBD
Baseline	73.7	77.2	70.8	74.9
+ ST	77.3	78.1	76.8	78.1
+ ST + IPT + FA	79.4	79.5	79.7	80.1

Table 7: Evaluation results of a small simulated life-long evolution sequence. Percentage of Cityscapes dataset represents different amounts of subset applied, whereas each of the percentage models is trained on the previous one with the corresponding Cityscapes subset.

# Images	None	25%	50%	75%	Full
mIoU	73.7	78.3	78.9	79.2	79.3

5 CONCLUSION

In this paper, we illustrate the necessity and importance of the interactive segmentation model to be capable of dealing with open scene images, which has been neglected by previous researchers. Specifically, we propose a life-long evolution framework for the interactive segmentation model, which allows it to evolve on unlabeled new target scenes by learning low-cost interactions provided by users. We conduct thorough experiments to show that with the help of our evolution framework, the interactive segmentation model manages to generalize without ground truth labels on different target datasets, and meanwhile remains a promising performance on previously witnessed scenes, demonstrating it as a practical way to interactively segment images from different domains within one single model.

ACKNOWLEDGEMENT

This work was supported in part by the Major Project for New Generation of AI (No.2018AAA0100400), the National Natural Science Foundation of China (No. 61836014, No. U21B2042, No. 62072457, No. 62006231).

REFERENCES

[1] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. 2020. BlendMask: Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8573–8581.

[2] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. 2016. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3640–3649.

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 801–818.

[4] Minghao Chen, Hongyang Xue, and Deng Cai. 2019. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2090–2099.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.

[6] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. 2021. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1601–1610.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 2 (2010), 303–338.

[9] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3146–3154.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

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

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

[14] Jianbo Jiao, Yunchao Wei, Zequn Jie, Honghui Shi, Rynson WH Lau, and Thomas S Huang. 2019. Geometry-aware distillation for indoor semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2869–2878.

[15] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. 2020. Continuous adaptation for interactive object segmentation by learning from corrections. In *European Conference on Computer Vision*. Springer, 579–596.

[16] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. 2017. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2359–2367.

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

[18] Long Ang Lim and Hacer Yalim Keles. 2020. Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications* 23, 3 (2020), 1369–1380.

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[21] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. 2020. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13339–13348.

[22] Nian Liu and Junwei Han. 2016. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 678–686.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[24] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. PMLR, 97–105.

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

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

[27] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 2014. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 891–898.

[28] Amed Mvoulana, Rostom Kachouri, and Mohamed Akil. 2019. Fully automated method for glaucoma screening using robust optic nerve head detection and unsupervised segmentation based cup-to-disc ratio computation in retinal fundus images. *Computerized Medical Imaging and Graphics* 77 (2019), 101643.

[29] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*. 1520–1528.

[30] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. 2020. Domain adaptive semantic segmentation using weak labels. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer, 571–587.

[31] Mark Pauly, Richard Keiser, and Markus Gross. 2003. Multi-scale feature extraction on point-sampled surfaces. In *Computer graphics forum*, Vol. 22. Wiley Online Library, 281–289.

[32] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. 2017. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924* (2017).

[33] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for data: Ground truth from computer games. In *European conference on computer vision*. Springer, 102–118.

[34] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3234–3243.

[35] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2012. Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* 23 (2012), 3.

[36] Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175* (2017).

[37] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. 2021. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14454–14463.

[38] Qi Wang, Junyu Gao, and Xuelong Li. 2019. Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. *IEEE Transactions on Image Processing* 28, 9 (2019), 4376–4386.

[39] Yuxi Wang, Jian Liang, and Zhaoxiang Zhang. 2021. Give me your trained model: Domain adaptive semantic segmentation without source data. *arXiv preprint arXiv:2106.11653* (2021).

[40] Yuxi Wang, Junran Peng, and ZhaoXiang Zhang. 2021. Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9092–9101.

[41] Yongji Wang, Qingwen Qi, Ying Liu, Lili Jiang, and Jun Wang. 2019. Unsupervised segmentation parameter selection using the local spatial statistics for remote sensing image segmentation. *International Journal of Applied Earth Observation and Geoinformation* 81 (2019), 98–109.

[42] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S Huang, and Honghui Shi. 2020. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12635–12644.

[43] John Winn and Nebojsa Jojic. 2005. Locus: Learning object classes with unsupervised segmentation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 1. IEEE, 756–763.

[44] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. 2017. Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243* (2017).

[45] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. 2016. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 373–381.

[46] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. 2021. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12414–12424.

[47] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. 2020. Interactive object segmentation with inside-outside guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12234–12244.

[48] Ervine Zheng, Qi Yu, Rui Li, Pengcheng Shi, and Anne Haake. 2021. A Continual Learning Framework for Uncertainty-Aware Interactive Image Segmentation. In

Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 6030–6038.

[49] Yi Zhou, Xiaodong He, Lei Huang, Li Liu, Fan Zhu, Shanshan Cui, and Ling Shao. 2019. Collaborative learning of semi-supervised segmentation and classification for medical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2079–2088.

[50] Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning* 3, 1 (2009), 1–130.

[51] Xiaojin Jerry Zhu. 2005. Semi-supervised learning literature survey. (2005).