Mining Semantic Correlations between Mispredictions and Corrections for Interactive Semantic Segmentation

Yutong Gao, Congyan Lang, Fayao Liu, Chuansheng Foo, Yuanzhouhan Cao, Lijuan Sun, Yunchao Wei

***Abstract*—Interactive semantic segmentation pursues high-quality segmentation results at the cost of a small number of user clicks. It is attracting more and more research attention for its convenience in labeling semantic pixel-level data. Existing interactive segmentation methods often pursue higher interaction efficiency by mining the latent information of user clicks or exploring efficient interaction manners. However, these works neglect to explicitly exploit the semantic correlations between user corrections and model mispredictions, thus suffering from two flaws. Firstly, similar prediction errors frequently occur in actual use, causing users to repeatedly correct them. Secondly, the interaction difficulty of different semantic classes varies across images, but existing models use monotonic parameters for all images which lack semantic pertinence. Therefore, in this paper, we explore the semantic correlations existing in corrections and mispredictions by proposing a simple yet effective online learning solution to the above problems, named Correction-Misprediction Correlation Mining (CM^2). Specifically, we leverage the correction-misprediction similarities to design a Confusion Memory Module (CMM) for automatic correction when similar prediction errors reappear. Furthermore, we measure the semantic interaction difficulty by counting the correction-misprediction pairs and design a Challenge Adaptive Convolutional Layer (CACL), which can adaptively switch different parameters according to interaction difficulties to better segment the challenging classes. Our method requires no extra training besides the online learning process and can effectively improve interaction efficiency. Our proposed CM^2 achieves state-of-the-art results on 3 public semantic segmentation benchmarks.**

***Index Terms*—Interactive image segmentation, Semantic image segmentation.**

I. Introduction

In the past few years, semantic segmentation has attracted much attention for its wide application scenarios, such as human behavior and attribute analysis [1]–[6], automatic driving [7]–[11], medical diagnosis [12]–[15], etc. Excellent semantic segmentation models [16]–[29] are often hungry for a large number of finely annotated pixel-level labels for training. More annotated data tend to bring better performance. However, the amount of publicly available semantic segmentation datasets is limited, and constructing more fine-grained training data is both labor-intensive and time-consuming. Therefore, interactive segmentation methods are gaining more attention, aiming to construct high-quality annotations quickly and conveniently with a small number of user-provided prior cues (labeled clicks, bounding boxes, *etc.*). The annotator improves the quality of the segmentation mask through iterative interaction with the model until a satisfactory result is obtained. After training, the model can be used for fast annotations with a few pixel-level clicks provided by users.

Existing interactive segmentation methods [30]–[47] often pursue the higher interaction efficiency by mining the latent information of user clicks or exploring efficient interaction forms. Recently, Gao *et al.* [48] explored the feasibility of semantic-level interactive segmentation and demonstrated its high efficiency compared to the object-level ones on human parsing task. Theodora *et al.* [49] proposed to use clicks as a constraint to optimize the model parameters during testing in an online learning manner to improve interaction efficiency. Though proved effective, these works neglect to explicitly exploit the semantic correlation hidden in the user corrections and the model mispredictions, which still suffer from two flaws: 1. In practical applications, similar prediction errors occur frequently, *e.g. "Hat" is often misclassified as "Hair"*, causing annotators to repeatedly correct them. 2. There is usually a particular semantic class that requires more user clicks / corrections, which varies across images. However, existing models use monotonous parameters when processing them, *i.e.*, processing all images with invariant semantic-agnostic convolution kernels, which lacks semantic pertinence and limits the model's capability of handling challenging classes.

Therefore, in this paper, we explore the semantic correlation between the user corrections and model mispredictions while proposing a simple yet effective online learning solution to the aforementioned problems, named **Correction-Misprediction Correlation Mining (CM^2)**. During the interaction, each correction click discloses a pixel with semantic perception difference between the model and the user. To better exploit similar perception differences, we categorize a correction-misprediction pair based on all the possible combinations,

Yutong Gao is with the Key Laboratory of Big Data & Artificial Intelligence in Transportation (Ministry of Education), School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China, Ministry of Education Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance, School of Information Engineering, Minzu University of China, Beijing 100081, China. Congyan Lang *(corresponding author)* is with the Beijing Key Laboratory of Traffic Data Analysis and Mining, School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China. Fayao Liu is a research scientist at Institute for Infocomm Research, A*STAR, Singapore. Chuansheng Foo currently leads a research group at the Institute for Infocomm Research, A*STAR, Singapore. Yuanzhouhan Cao is currently a lecturer at the School of Computer Science and Information Technology, Beijing Jiaotong University, Beijing 100044, China. Lijuan Sun is currently an associate researcher in the Intellectual Property Information Service Center, Beihang University, Beijing 100191, China. Yunchao Wei is with the Institute of Information Science, Beijing Jiaotong University, Beijing, China, 10044. E-mail:(wychao1987@gmail.com).

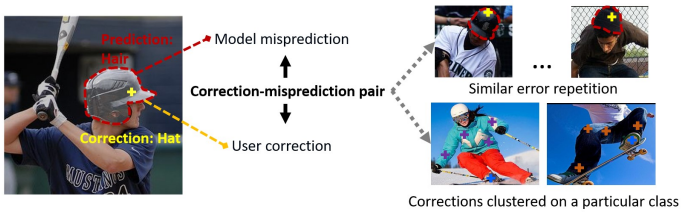


Fig. 1. cognitive biases diagram. Different colored "+" symbols represent different classes of correction clicks.

termed as confusion category.

Furthermore, we observe the correction-misprediction pairs exhibit two characteristics (refer to Section IV for relevant statistics), as shown in Fig. 1. First, corrections of a specific class tend to repeat across several confusion classes, *i.e.*, correction-misprediction pairs have strong semantic correlations, which makes their features regular to follow. Second, corrections / clicks tend to cluster together in a particular class, which indicates that there is often a semantic class with a prominent amount of correction-misprediction pairs that is harder to segment in an image. We refer this class as the semantic challenging class (SClg-cls). Based on the former, we design a **Confusion Memory Module (CMM)** to reduce the errors that have seen before according to previous correction-misprediction pairs. Specifically, we construct an online updated memory bank for the correction-misprediction pairs. Whenever similar prediction errors reappear, the model can automatically correct them according to the memorized knowledge. The features stored in the memory bank are classified according to the confusion category, *i.e.*, correction-misprediction pairs, and those features that are not used frequently are regularly forgotten. At the same time, to maintain the feature consistency in the memory bank, we generate these features by adopting the momentum parameter updating method [50]. To tackle the second issue, we propose a **Challenge Adaptive Convolutional Layer (CACL)**, which aims to enhance the model's capability of handling the SClg-cls. Specifically, we design the convolutional layer with multiple switchable convolution parameters for different classes. It can be inferred that the class with more corrections is harder to be segmented / corrected, hence we leverage the correction click numbers to measure how challenging a class is (semantic challenge degree), as well as do the parameter switching in CACL. We treat the class with the most amount of correction-misprediction pairs as the SClg-cls, and load the corresponding parameters for feature processing, thereby improving the model's capability of segmenting the SClg-cls and reducing the interference of other classes during learning. Considering the possible biases of the semantic challenge degree measurement, we construct a categorical distribution based on correction click numbers and sample the SClg-cls, to which the convolution parameters are switched for robustness. An overview of our model is shown in Fig. 2(a).

We conduct extensive experiments on LIP [51], Cityscapes [52] and Pascal-Person-part [53] datasets. We demonstrate that both the proposed CMM and CACL can effectively improve the efficiency of interactive semantic segmentation. It is worth mentioning that, based on the online learning [49] method, our CMM and CACL can benefit the interaction efficiency without additional training, and save efforts of retraining the model from beginning. Our method achieves state-of-the-art results on all the datasets. Furthermore, our method is equally applicable to interactive object segmentation, which we demonstrate on the Berkeley [54] dataset.

In conclusion, the major contributions of this paper lie in the following aspects:

- We examine the semantic correlation between user corrections and model mispredictions, as well as prove its importance for the interactive segmentation task through rich experiments.
- We design a simple and effective online learning scheme, *i.e.*, **Correction-Misprediction Correlation Mining ($\mathbf{CM^2}$)**, for interactive semantic segmentation. It can effectively improve the interaction efficiency by reducing similar mispredictions and enhancing the model's capability of handling challenging classes.
- Our CM^2 achieves state-of-the-art performance and can **save ∼20%, 25% and 30.76% of correction efforts** on LIP (to achieve 85% mIoU), Cityscapes (to achieve 89% mIoU) and PASCAL-Person-Part (to achieve 85% mIoU) datasets.

II. Related Works

Interactive segmentation aims to quickly refine the segmentation mask with the guidance of annotator interactions, which makes it suitable for relieving human efforts when labeling pixel-level annotations.

Efforts are made to explore efficient ways to improve the interaction efficiency for object-level interactive segmentation. Their focuses can be roughly divided into three categories, *i.e.*, rational click simulation strategies [30]–[38], efficient click transformation methods [30], [31], [39]–[42] and training procedure refinement [42]–[47]. For click simulation strategies, the existing literature is split between clicks inside the object/background regions [33], [42]–[47], [55] or clicks at the object border [30], [31], [34]–[36]. [55] proposes to interact at the center area of objects/mislabelled regions. [33] suggests locating the object by one inside point and two outside point for reducing interaction effort. [30] proposes using extreme clicks on edges to locate objects. [31] uses click dragging to strengthen the model's capability of error correction. For click transformation, several works [30], [31] encode clicks as Gaussians, which can encode all the clicks in a single channel. In [39], [40], the authors investigate the use of super-pixels to generate the localization maps by providing guidance with local consistency. [41], [42] transform the clicks based on the Euclidean distance calculation, leveraging extracted features to enhance the transformation expression. For training procedure refinement, RIS-Net [46] adds a local branch to refine the predicted result around annotator clicks. f-BRS [43] refines the intermediate features to get more precise masks with a faster speed compared with BRS [41]. [47] predicts multiple potential results and trains another network to choose from them. FCANet [44] underlines the importance of the first click

and proposes first click attention to get better results. [42] propagates labeled representations from clicks to conditioned destinations to better exploit user cues. In particular, Theodora *et al.* [49] propose to use clicks as a constraint to optimize the model parameters during testing in an online learning fashion.

Gao *et al.* [48] explore the feasibility of semantic-level interactive segmentation and demonstrate its high efficiency compared to the object-level ones, interactive semantic segmentation can achieve higher interaction efficiency and alleviate user labeling efforts in specific tasks such as human parsing and scene parsing. We therefore focus and rely on semantic-level interactive segmentation task to do the investigation. Compared with the object-level interactive segmentations, the advantage of semantic-level interactive segmentation is that the clicks of different semantic classes can fully complement their semantic cues with each other, and efficiently guide the data annotation of semantic segmentation tasks. On the contrary, most object-level interactive segmentation methods need to annotate each semantic class separately during the annotation process, which makes it difficult to share the semantic information of different classes and leads to inefficient interactions.

Though proved effective, existing interactive approaches ignore the semantic correlations between user correction and model mispredictions, which raises at least two problems: 1. In practical applications, same correction-misprediction pairs appear frequently, *e.g.*, "Hat" is often misclassified as "Hair", causing annotators to repeatedly correct similar prediction errors. 2. The interaction difficulty (manifested by the number of correction-misprediction pairs) of different classes varies across images, but existing models use monotonic parameters for them all, which lacks semantic pertinence and limits the model's capability of efficient interactions with challenging classes. Therefore, this paper explores an interactive online learning solution that mines these correction-misprediction correlations to improve the interaction efficiency. We hope this could inspire follow-up researches.

III. Proposed Method

In this section, we first introduce the basic interactive semantic segmentation process we employ, then describe the definitions and characteristics of the correction-misprediction pairs, as well as the measurement of semantic challenge degree for each category, and finally we describe the network construction architecture of our CM^2 in detail.

A. Interaction Process

We adopt the interactive semantic segmentation process proposed by Gao *et al.* [48], which is shown to be reasonable and efficient. The process is divided into two stages, namely the initialization stage and the correction stage. In the initialization stage, the central area of each object / part is given a click to get the initialization mask. In the correction stage, a single click is added to the largest mislabeled area in each round of interaction until a satisfactory result is generated. As mentioned by Gao *et al.*, the correction stage is relatively time-consuming because the annotator needs to iteratively look for mislabeled regions, which is what this paper focuses on.

B. Semantic Correlations Disclosed by Correction-misprediction Pairs

Here, we describe in detail the definitions and characteristics of the correction-misprediction pairs, as well as the measurement of semantic challenge degree for each category.

Correction-misprediction pairs are essentially disclosures of the differences between the annotator and the model in semantic perception of pixels. In our quest to uncover the influence of semantic correlation between mispredicted pixels and correction clicks on interactive semantic segmentation, we introduce C to represent the number of classes in the dataset. We formulate a **confusion category** $g^t \in \mathcal{G}$ where $\mathcal{G} \subseteq 1, ..., C \times 1, ..., C$, viewed as a pair of distinct semantic classes, for instance, (i, j). This suggests in the t-th round of interaction, the model prediction is i while the annotator correction stands at j ($i \neq j$). Evidently, we have $|\mathcal{G}| = C \times (C-1)$, implying each mispredicted class can pair with a correction class other than itself to establish a confusion category, enabling coverage of all perceivable combinations of mispredicted and correction semantic classes. A confusion category represents the perception difference at the clicked pixel. However, the features of a single pixel have limited capability of representing the confusion category. Therefore for each g^t, we construct a confusion prototype to better characterize different confusion categories. Specifically, for image I, let R^t be its semantic prediction after t-th round interaction. Then during $t+1$ round interaction, a single click is added to the largest mislabeled region, which results in a confusion category g^{t+1}. The model is then updated accordingly to produce prediction R^{t+1}. We then construct a set of pixels $\mathcal{S}$ whose prediction changes from round t to $t+1$ are consistent with g^{t+1}:

$$\forall p \in I, \quad p \in \mathcal{S} \quad \text{if} \quad (R_p^t, R_p^{t+1}) = g^{t+1}. \tag{1}$$

Here R_p^t represents the predicted class of pixel p in R^t. Then we calculate the average feature of $\mathcal{S}$ in round t as a **confusion prototype** of g^{t+1}. To better capture the characteristics of the confusion categories, we generate and record multiple confusion prototypes for each confusion category, and design a CMM to correct similar prediction errors (refer to Section III-C for details).

Correction-misprediction pairs can also reflect the challenge degree of segmenting different semantic classes. We use the number of correction clicks to measure the challenge degree of each semantic class. A class with more correction clicks is more challenging to segment. At the same time, we observe correction clicks tend to cluster in a particular semantic class per image (refer to Section IV). We can infer that there exist a class whose semantic challenge degree is significantly higher than other classes in most images. Therefore, we consider this semantic class as the SClg-cls which the model should focus on during learning and processing. Note that SClg-cls varies for different images. Thus, instead of using a traditional convolutional layer, we design a CACL to better learn and handle the challenging class by switching the semantic-aware convolutional parameters corresponding to the SClg-cls (refer to Section III-C for details).

C. Network Architecture

Network Overview. We adopt the DeepLab V3+ [56] as the base segmentation network, which is suitable for interactive image segmentation in multiple previous works [33], [43], [45], [48], [49]. We adopt the method proposed by Theodora *et al.* [49], which is using clicks as labels for online learning. It can simply and effectively improve the interaction efficiency. In addition, in practical applications, there is a certain domain gap between the images to be labeled (test images) and the training set images, the algorithm based on online learning can effectively alleviate it. We adopt the online learning loss function in the CA-learning [49], which can be expressed as:

$$loss = \lambda \mathcal{L}_{CE}(pred, init) + (1-\lambda)\mathcal{L}_{CE}(pred, corr) + \gamma \mathcal{L}_F, \tag{2}$$

where $\mathcal{L}_{CE}(gt)$ is the standard cross-entropy loss compute with label gt, $init$ is the initialization mask, $pred$ is the segmentation result of the model, $corr$ is a sparse label only consist of correction clicks, λ is a hyperparameter that controls the weight between $\mathcal{L}_{CE}(init)$ and $\mathcal{L}_{CE}(corr)$. $\mathcal{L}_F$ is a cost for changing network parameters and γ defines the strength of parameter regularization. $\mathcal{L}_F$ can be formulated as

$$\mathcal{L}_F = \Omega^T(\varphi - \varphi^*)^{\bigodot 2}, \tag{3}$$

where φ^* are initial model parameters before update in each round of interaction, φ are the updated parameters and Ω is the importance of each parameter which generated by the Memory-Aware Synapses (MAS) computation. $(.)^{\bigodot 2}$ is the element-wise square (Hadamard square). Details refer to [49].

We present an overview of our model in Fig. 2(a). Like most interactive segmentation methods, we feed the transformed clicks together with the image into the network for feature extraction, please refer to Section IV for training details, *e.g.*, click simulation. To break the limitation of monotonic parameters and better deal with the SClg-cls, we set a CACL instead of the feature fusion layer of ASPP [17] module, and switch the convolution parameters according to the correction click number of each semantic class. In addition, to reduce the errors similar to the previously appeared confusion category, we feed the extracted features into CMM and build a memory bank to store the existing confusion prototypes. Image features are compared with the storage in the memory bank to identify those regions that may exist similar prediction errors, then a **confusion guidance matrix** is generated to guide feature optimizations.

Confusion Memory Module. To memorize the mispredictions and their corrections that have seen, we construct a memory bank to store confusion prototypes of each confusion category. As shown in Fig. 2b, we store the confusion prototypes separately in the memory bank, with each category memorizing up to 5 prototypes. During each round of interaction, those memorized prototypes with high similarity to image features are "activated". The memory bank is updated every s interaction rounds, and new memorized prototypes are generated to replace the least frequently activated prototypes in each category. In particular, in order to maintain the consistency of the confusion prototypes in the memory bank, we use the momentum strategy [50]. Let w_t be the model parameters at t-th online training step, then the momentum parameters w^m is calculated as:

$$w^m_{t+1} = \delta w^m_t + (1-\delta) w_{t+1}, \tag{4}$$

where w_0 is the parameters learned on the training set. δ is a hyperparameter that controls the magnitude of model parameter updates. We set δ to 0.999, and show in Section IV that this process has little effect on computation time.

We construct CMM to reduce the annotation efforts caused by the same confusion category, the structure of the CMM is shown in Fig. 3(a). We input the image feature F into the CMM and calculate the cosine similarity with the confusion prototypes of g stored in memory bank. For each $f \in F$, the similarity can be expressed as:

$$\text{simi}^g_f = \max_{u \in \{1,2,\ldots,5\}} \left(\frac{f \cdot (o^u_g)^T}{||f|| \cdot ||o^u_g||} \right), \tag{5}$$

where $o^u_g (u \in \{1,\ldots,5\})$ are the confusion prototypes of category g in the memory bank. The simi^g_f represents the similarity between the prototypes of g and the feature f. We use the maximum similarity within the same confusion category here. Please refer to Section IV for ablation studies on this design choice.

Then we retain those high similarities to construct the confusion guidance matrix X. The element of X is calculated as:

$$x^g_f = \begin{cases} 0, & \text{simi}^g_f < \theta \\ \text{simi}^g_f, & \text{simi}^g_f > \theta \end{cases}, g \in \mathcal{G}, f \in F \tag{6}$$

where x^g_f indicates the retained similarity of confusion category g for feature f. θ is a hyperparameter used to control the threshold of the minimum similarity retained. It is not difficult to see that the dimension of X is $[C \times (C-1), H, W]$, where C is the number of classes, H and W are the height and width of the feature map F. That is, each element $x \in X$ is represented by a $C \times (C-1)$ dimensional vector to record the possible confusion categories. Finally, we map the similarities in X to the feature space of F through a convolutional layer and fuse it with F, thereby promoting the model to reduce potential mispredictions. The fusion is formulated as:

$$F' = F + F \circ \text{Conv}(X), \tag{7}$$

where $\circ$ is the Hadamard product operation, the Conv is a dimension transform convolution layer with 1×1 kernel and the output dimension is the same to F. F' is the output feature of CMM. In order to adapt to the setting of online learning, we initialize $Conv$ with zeros.

Challenge Adaptive Convolutional Layer. In practice, SClg-cls vary from image to image, and interactions of SClg-cls are generally less efficient. Different from general image segmentation tasks, in interactive segmentation, we can measure the semantic challenge degree by the correction click numbers of different classes. This enables us to build and switch specific parameters for those classes with high interaction / segmentation difficulty, thereby reducing the interference of other classes on parameter learning.

The CACL structure is shown in Fig. 3(b). We construct a learnable convolution parameter list to store the parameters

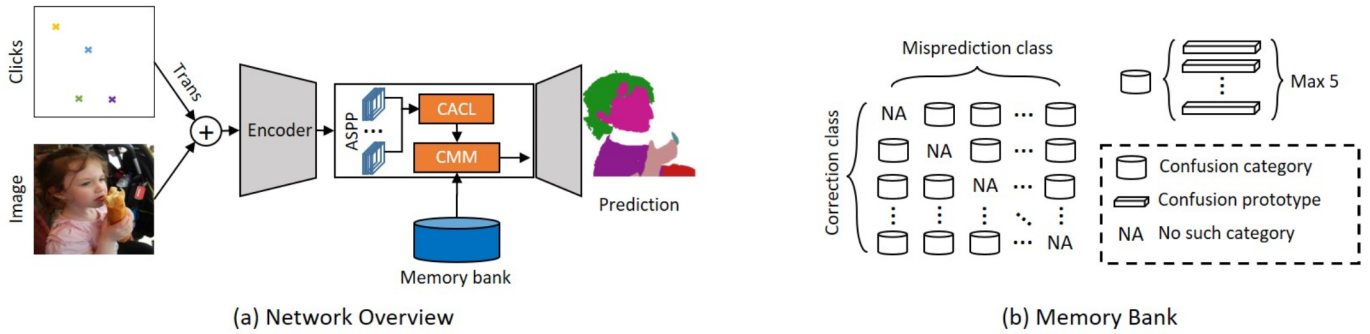


Fig. 2. (a) Network structure of our CM^2. A CACL is used instead of the feature fusion layer of ASPP [17] module for switching convolution parameters for SClg-cls. Then the features are fed into CMM to reduce the similar errors by constructing a memory bank of confusion prototypes. "Trans" represents click transformation [48]. (b) Memory bank structure diagram. Each confusion category consists of a pair of misprediction and correction classes, which confusion prototypes are stored according to.

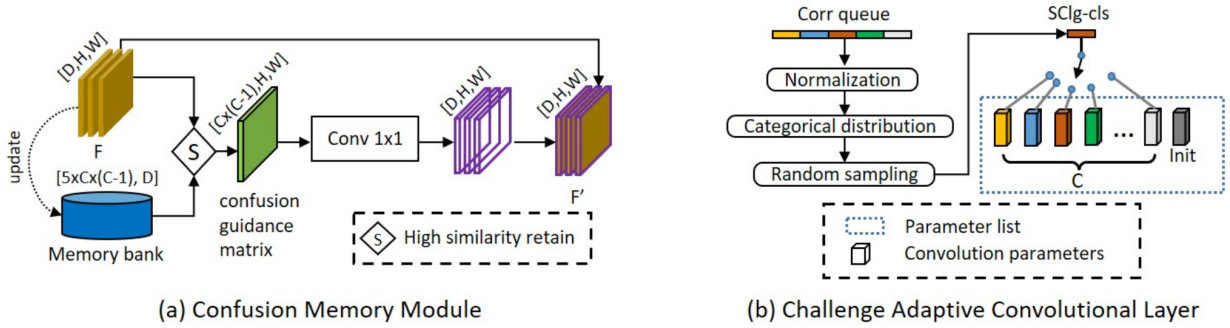


Fig. 3. (a) CMM illustration. "F" and "F′" indicate the input and output features respectively. "C" indicates the total class number. "D", "H" and "W" denote the dimension, hight and weight of features. The symbol "+" represents the concatenation of the location map and RGB channels, while different colored "x" symbols represent interaction clicks for different categories. (b) CACL illustration. "Corr queue" represents the queue of correction click numbers in different semantic classes. "C" indicates the total class number. "Init" represents the parameters for the initialization stage.

corresponding to each semantic class. In particular, we store a separate set of parameters for the initialization stage in which there are no correction clicks. In each round of interaction, we first count the correction click number of each class for locating the SClg-cls. Considering that the measurement based on correction click number may be biased in a few cases, *e.g. annotation errors, multiple challenging classes*, which cannot accurately reflect the semantic challenge degree. We hence sample the SClg-cls based on the categorical distribution constructed from correction click numbers, to add some randomness for parameter switching to make CACL more robust. Let N_c be the number of correction clicks for class c ($c \in \{1, \ldots, C\}$), then the SClg-cls $Mcls$ can be expressed as:

$$\text{Mcls} \sim \text{Cat}(P_1, \ldots, P_C)\ , \text{ with } P_c = \frac{N_c}{\sum_{i=1}^{C} N_i}\ ,\ c = 1, \ldots, C, \tag{8}$$

where $\text{Cat}(\cdot)$ denotes the categorical distribution of correction click numbers. The CACL loads parameters corresponding to class Mcls in the parameter list for the forward / backward propagation. In particular, in order to make the CACL directly applicable to the online learning process, we replace the feature fusion layer of ASPP with CACL and retain its parameters to initialize CACL.

IV. Experiments

In this section, we construct extensive experiments to validate our proposed CM^2 from multiple perspectives. We first present in turn the datasets we used, the implementation details, and the evaluation metrics. Then, we provide diverse ablation experiments to verify the effectiveness of each of our design choices. Then, we demonstrate the efficiency of our model by comparing with the existing powerful state-of-the-art interactive semantic segmentation methods. Finally, we also provide other experimental results to support our design choices and analyze the limitations of our model.

A. Datasets

The **LIP** dataset [51] is a large-scale benchmark for semantic segmentation of human images, consisting of 50,462 images with pixel-level annotations on 19 semantic body part labels. It includes a training set (30,462 images), a validation set (10,000 images) and a test set (10,000 images).

The **Cityscapes** dataset [52] is tasked for urban scene understanding, and contains 30 classes, 19 of which are used for scene parsing evaluation. It contains 5,000 pixel-level finely annotated images split into train/validation/test sets of 2,975/500/1,525 images.

The **PASCAL-Person-Part** dataset [53] provides pixel-level labels for six human body parts. There are 3,535 annotated

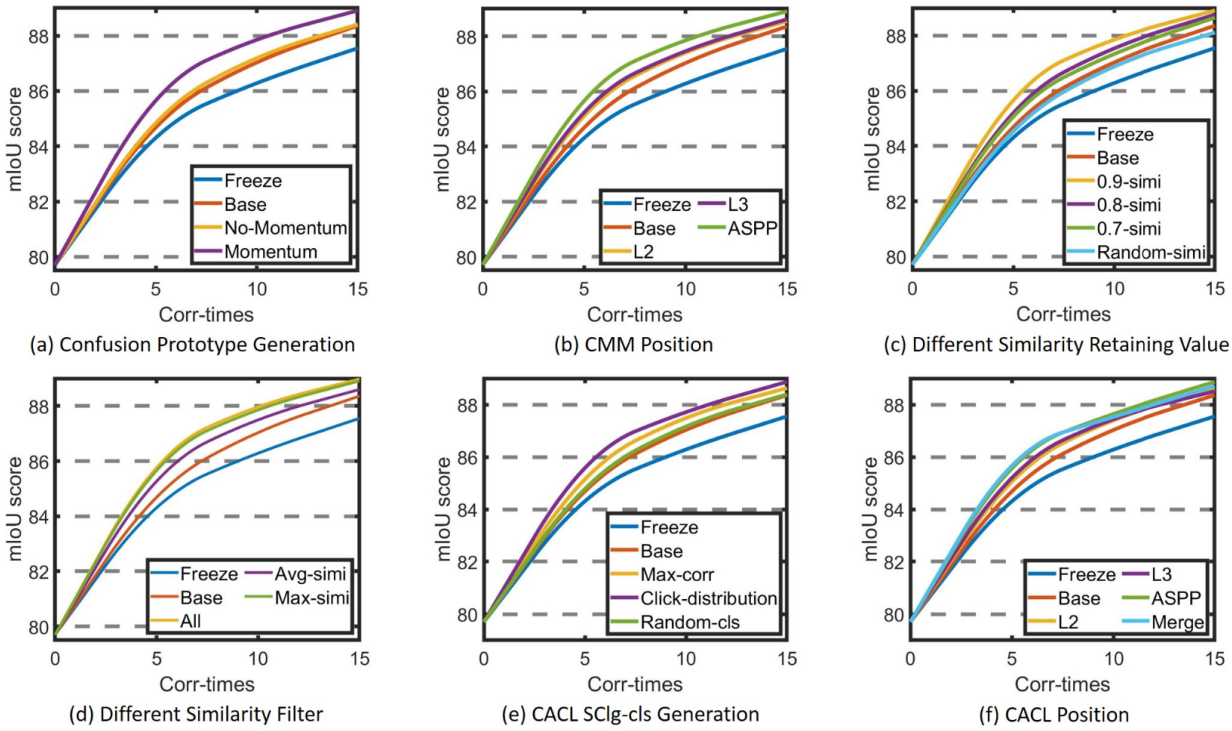


Fig. 4. Experimental results of CMM / CACL design choices. "corr-times" represents the correction click number. "Freeze" indicates the model without online learning setting. "Base" represents the baseline model. (a) Comparisons of different confusion prototype generation methods. "Momentum" and "No-Momentum" represent the generation of confusion prototype with or without using momentum parameter. (b) Comparisons of different positions for CMM. "L2", "L3" and "ASPP" indicate we add CMM after convolutional block 2, 3 and the ASPP module of the DeepLab V3+ [56], respectively. (c) Comparisons of different similarity thresholds for confusion guidance matrix. "0.9-simi", "0.8-simi", "0.7-simi", and "Random-simi" represent set θ. (d) Comparisons of different similarity filter for confusion guidance matrix generation. "Max-simi", "Avg-simi" and "All" represent filtering the confusion prototype similarities by taking the maximum value, the average value, and retaining all values in Eq. 5, respectively. (e) Comparisons of different design choices for SClg-cls generation. "Max-corr", "Click-distribution", and "Random-cls" represent select the SClg-cls by maximum correction numbers, probability distribution of correction numbers, and randomly sampling of the semantic categories, respectively. (f) Comparisons of different positions for CACL. "L2", "L3" and "ASPP" indicate we set the CACL as the last convolution layer of convolutional block 2, 3 and the ASPP module, respectively. "Merge" represents setting CACL by "L2","L3" and "ASPP" simultaneously.

images split into a training set of 1,717 images and a validation set of 1,818 images.

For all the datasets, we initialize model parameters with the training set and use the validation set for online learning and testing.

B. Implementation Details

Learning Settings. We adopt the DeepLab V3+ [56] as the base segmentation network and pre-train the initial parameters on the training set. The training settings and click-transform method are consistent with CM [48]. Following the CM [48] in training, we simulate central clicks and extra clicks for locating semantic parts and cover various clicks respectively. We describe the simulation method in detail below. During testing, we adopt the online learning method proposed by CA-Learning [49] to update the model, we set λ to 0.7 and γ to 0.2. We set the learning rates of "single image" and "image sequence" in the CA-learning to 1e-6 and 1e-7 respectively, we employ the result as a strong baseline. Our CM^2 is designed for online learning, and the proposed CACL and CMM do not require any learning on the training set. We set θ and s in CMM to 0.9 and 30, respectively. Considering the influence of the testing image order on the online learning process, we read the images randomly and calculate the mean of 3 experimental results as the final result.

Click Simulation. Here we describe our simulation method in detail. We simulate central clicks and extra clicks for locating semantic parts and cover various clicks respectively. For central clicks, we adopt the Central-Click simulation strategy [48]. We simulate a central click roughly in the center area of each connected semantic class region. We randomly sample 5 points as the candidate set for a central click, then we select the point which has the largest Euclidean distance away from the class boundary as the central click. For extra clicks, we refer to the iterative training method [45] for simulation. We loop our model two times for each batch during training, we generate an initial prediction in loop one by the central clicks and randomly pick 15 points from the mislabeled pixels as the extra clicks for each image. Then in loop two we feed all the simulation clicks into the model for the final prediction.

C. Evaluation Metrics

Following the semantic-level interactive works [48], we use the Mean Intersection Over Union (mIoU) and the click number as the evaluation metrics. Specifically, we perform different number of correction clicks on each image to observe

the change of mIoU, better interaction efficiency c an be reflected i n t wo a spects: 1 . A chieve h igher m IoU w ith the same number of clicks, 2. Achieve a preset mIoU with fewer clicks.

D. Ablation Study

We conduct ablation experiments on the LIP validation set to verify the effectiveness of the CM^2 design, including the effect of CMM and the effect of CACL. To investigate the effect of CMM, we separately construct variants for the generation of confusion prototypes in memory bank, the CMM position in the network, and the design choices of the confusion guidance matrix. To explore the effect of CACL on interaction efficiency, we separately construct variants for the generation of SClg-cls and the position of CACL in the network. We also visualize the optimization effect of CMM and CACL on several LIP / Cityscapes images.

Besides, we provide more experimental results including detailed statistics for the cluster of corrections in the SClg-cls, and the correction frequency with the same confusion category.

Effectiveness of the Momentum Based Confusion Prototype Generation. To investigate the effectiveness of the confusion prototype generation method for the memory bank, we construct two variants to compare with the baseline, which are the generation methods with and without the momentum parameters, named "Momentum" and "No-Momentum" respectively. As shown in Fig. 4(a), using the momentum parameters keeps the confusion prototypes in memory bank consistent during online learning and improves mIoU performance by about 1.1%. In contrast, the variant without using the momentum parameters does not gain significant performance improvement. We analyze that maintaining the representational consistency of the confusion prototypes is necessary for CMM learning, and the momentum-based prototype generation method can effectively reduce the representational discrepancies caused by parameter updating, thus providing a more reliable confusion guidance matrix to assist CMM in locating potential error areas.

Effects of Different CMM Position. To explore how the CMM position in the network effect the performance, we construct three variants by adding CMM after the Layer2, Layer3, and ASPP module of the DeepLab V3+. The results are shown in Fig. 4(b), the variant add CMM after the ASPP module shows the best performance and improve the mIoU score about 1.1% compare to the baseline model, while other variants slightly bring around 0.33% improvements, which indicates that deep features are more suitable for CMM than the lower ones.

Impact of Mispredicted Features on the Confusion Prototype. During the computation of the confusion prototype, it is inevitable that some mistakenly predicted pixel features participate in the calculation of the confusion prototype. Here, we investigate the influence of such mispredicted pixels on the confusion prototype and the model performance. First, we calculate the proportion of mispredicted pixels used in the confusion prototype calculation. Specifically, by comparing with the ground truth labels during testing, we identify the pixels that are mispredicted during the calculation of the confusion prototype. As shown in Tab. I, the portion of mispredicted pixels increases with the number of interaction iterations, ranging from an initial 11.68% to a peak of 19.77%, never surpassing 20%. This initially leads us to conclude that mispredicted pixels have but a limited effect on the confusion prototype.

Furthermore, we compared the confusion prototypes calculated with and without mispredicted pixel features to observe the specific impact of these mispredictions. Specifically, mispredicted pixel features are removed from the confusion prototype calculation generating a "noise-free" confusion prototype, which is then compared with the confusion prototype from our original method in term of cosine similarity (average on the test set). Results, indicated in Tab.I, demonstrate consistently high similarity levels between the two prototypes. It is remarkable to see that, even when the proportion of mispredicted pixels nears 20%, the similarity between the two confusion prototypes still reaches 98.02%. This signifies the minimal influence of mispredicted features on the confusion prototype. We posit that this is due to the model's unifying representation of the confusion prototype, even if it contains inaccuracies.

Lastly, we construct a variant model that employs noise-free confusion prototypes, and compare its performance to our original method, as shown in Fig. 6(a). It is clear that the mIoU performance of both methods is very similar, confirming that mispredicted features exert negligible impact on model performance in the computation of the confusion prototype.

Investigation of Different Similarity Retention Threshold for Confusion Guidance Matrix. To prove the effectiveness of the high similarity retention for the confusion guidance matrix in CMM, we construct four variants by retaining the similarity value higher than different thresholds, named "0.9-simi", "0.8-simi", "0.7-simi", and "Random-simi", generating the confusion guidance matrix by setting the θ to 0.9, 0.8, 0.7, and a randomly threshold between 0 and 1 for each image, respectively. As shown in Fig. 4(c), the 0.9-simi variant performe better than the other variants, and the higher similarity retention tend to bring more mIoU improvement, the Random-simi variant even trigger mIoU decrease. This is mainly because low similarity retention introduce more noise in the confusion guidance matrix and reducing the signal-to-noise ratio, limits the CMM capability to locate error regions. The results indicates that the confusion prototypes are regular to follow, and our CMM is more suitable for leveraging the high similarity between the confusion prototypes and the features to guide the interaction.

Effects of Different Similarity Filter Method for Confusion Guidance Matrix. For CMM, we investigate different design choices of confusion guidance matrix generation, and explore which similarity filter method can better mine the potential confusion category areas. Specifically, we build three variants of similarity calculation methods between confusion prototype and image feature: 1. Keep the similarity of all the five confusion prototypes, named "All", 2. Retain the maximum similarity for each confusion category, named "Max-simi", 3. Keep the average similarity of each confusion category, named

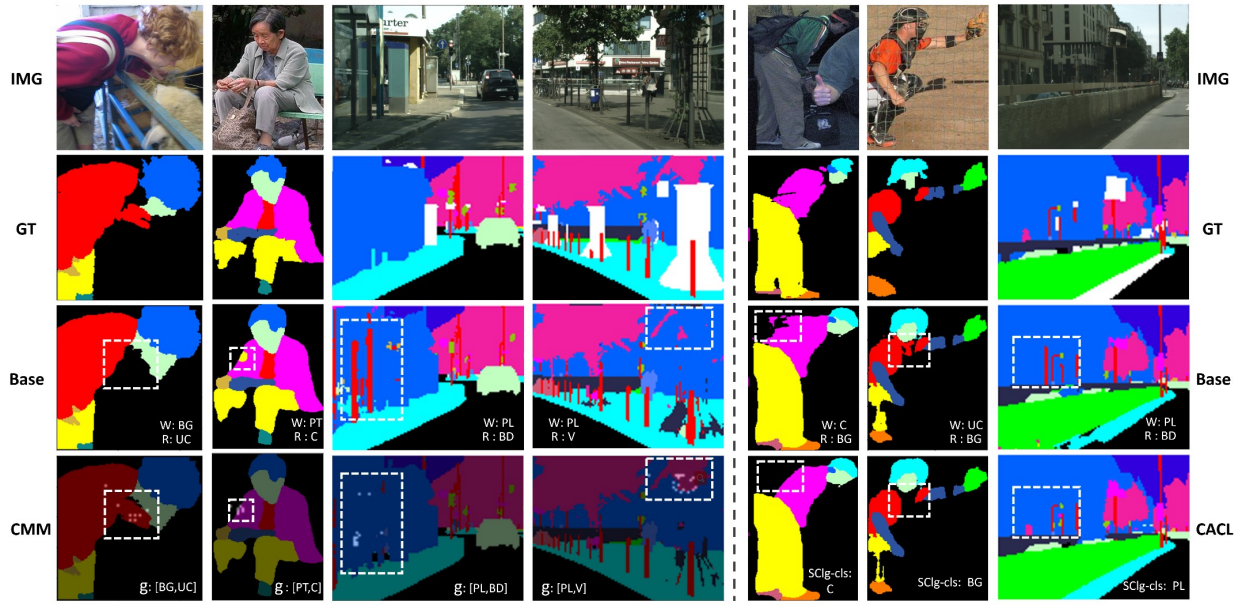
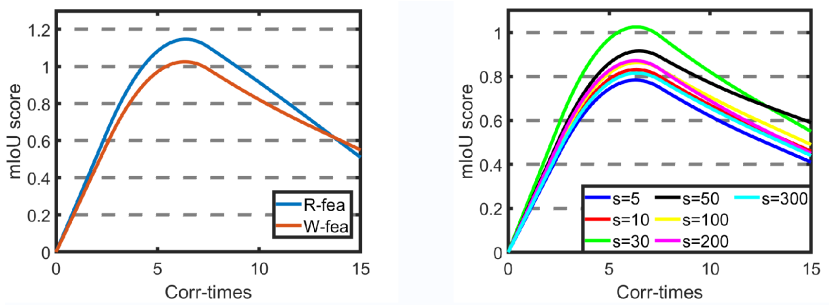


Fig. 5. Effectiveness visualization of CMM (left) and CACL (right) on LIP / Cityscapes datasets. The light areas of CMM results are the high similarity regions in the confusion guidance matrix. "Base" represents the results of our baseline model. "W" and "R" represents the wrongly predicted label and the right annotated label. "BG", "UC", "C" and "PT" indicate the background, upper cloth, coat and pants classes of LIP dataset, respectively. "PL", "BD" and "V" indicate the pole, building and vegetation classes of Cityscapes dataset.



(a) Impact of misprediction pixels on the confusion prototype in CMM

(b) Effects of different update frequencies of memory banks

Fig. 6. (a) Impact of mispredicted pixel features in the confusion prototype calculation on model performance (LIP dataset). "W-fea"/"R-fea" respectively denote the confusion prototype calculation when it includes/excludes mispredicted pixel features. (b) Influence of the update frequency in the memory bank of the CMM module on performance. "s" represents the hyperparameter s controlling the update frequency of the memory bank.

TABLE I

IMPACT OF MISPREDICTED PIXELS ON CONFUSION PROTOTYPE (LIP DATASET). "W-PCT" REPRESENTS THE PROPORTION OF MISPREDICTED PIXELS USED IN THE CONFUSION PROTOTYPE CALCULATION. "ITERATION" REPRESENTS THE INTERACTION ITERATION ROUND. "WR-CS" DENOTES THE AVERAGE COSINE SIMILARITY BETWEEN THE CONFUSION PROTOTYPE EXCLUDING AND INCLUDING THE FEATURES OF MISPREDICTED PIXELS.

	Iteration 1	Iteration 2	Iteration 3	Iteration 5	Iteration 10	Iteration 15
W-pct(%)	11.68	13.84	13.89	14.46	16.94	19.77
WR-cs	99.02	98.67	98.63	98.24	98.06	98.02

"Avg-simi". The "All" variant generate the confusion guidance matrix five times larger than the other two variants. The experimental results on LIP [51] validation set are shown in Fig. 4(d). The "All" and "Max-simi" variants perform similar effectiveness, while "Avg-simi" variant is slightly inferior to the other two. This indicates that keep all the similarities can provide the most complement clues in confusion guidance matrix, and the Max-simi variant can retains more valuable clues than the Avg-simi variant when the prototypes of a confusion category have large difference, the Avg-simi variant perhaps filtering out some useful clues. Therefore we adopt the Max-simi variant in our paper for less memory consumption.

Impact of the Update Frequency of the Memory Bank. We investigated the performance impact of the update frequency of the memory bank in CMM by manipulating the hyperparameter s and thereby creating models exhibiting different update frequencies. s is set at five different values, *i.e.*, 5, 10, 30, 50, 100, 200, and 300, while the corresponding shifts in mIoU are monitored. Fig. 6(b) illustrates that the model performance peaks when s = 30. A too-high update frequency may inadequately reflect the activation probability of confusion prototypes in the memory bank and risks overwriting beneficial features. Conversely, a memory bank updated too infrequently may not efficiently capture valuable

TABLE II
QUANTITATIVE COMPARISON TO STATE-OF-THE-ART INTERACTIVE SEMANTIC SEGMENTATION METHODS ON THREE BENCHMARKS. "AVG." AND "CORR." REPRESENTS THE AVERAGE CLICK NUMBER ON CUMULATIVE CATEGORY AMOUNT AND THE CORRECTION CLICK NUMBER OF EACH IMAGE FOR ACHIEVING THE PRESET MIOU SCORE. "PASCAL" IS SHORT FOR THE PASCAL-PERSON-PART DATASET.

	LIP 85% mIoU		Cityscapes 89% mIoU		PASCAL 80% mIoU		PASCAL 85% mIoU	
	Corr.	Avg.	Corr.	Avg.	Corr.	Avg.	Corr.	Avg.
ITIS [45]	6	1.82	15	3.64	5	2.86	-	-
CM(N) [48]	9	2.19	-	-	15	4.46	-	-
CM(R) [48]	8	2.07	-	-	10	3.66	-	-
CM(SP) [48]	7	1.95	-	-	6	3.02	-	-
CA-learning [49]	5	1.7	12	3.40	4	2.7	13	4.14
CM^2(ours)	4	1.58	9	3.16	3	2.54	9	3.5

confusion prototypes, thereby impairing the CMM's capability to correct similar misprediction.

Visualization Result of CMM. The CMM visualization results are shown in the left part of Fig. 5. We can clearly observe that the prediction results generated with the same clicks are improved by using CMM over the baseline in high similarity regions of the confusion guidance matrix. Which shows the effectiveness of our CMM in locating the potential mislabeled area.

Effects of Different SClg-cls Generation Methods. For the generation of SClg-cls in CACL, we construct three variants to compare with the baseline: 1. Directly select the class with the maximum correction numbers, called "Max-corr", 2. Sample the SClg-cls from probability distribution of correction click numbers, called "Click-distribution", 3. Randomly sample the SClg-cls, called "Random-cls". As shown in Fig. 4(e), the Random-cls variant performance is less efficient and its performance is close to the CA-learning, the Max-corr variant can improve the mIoU performance by about 0.6%, and the Click-distribution variant expands the performance gain to about 1%. This indicates that the parameter switching for SClg-cls can effectively improve the interaction efficiency, and adding randomness to correction click numbers makes the model more robust to cases where SClg-cls inaccurately reflect semantic challenge degree.

Effects of Different CACL Position. For the CACL position in the network, we set the last convolution of Layer 2, Layer 3, and the feature fusion layer of ASPP as CACL, plus the combination of all the three positions to construct a total of 4 variants. The results are shown in Fig. 4(f). The variants of ASPP feature fusion layer and the combination layers have better performance, which indicates that CACL is more suitable for handling deep features than low-level features, and the semantic challenge degree are more evident in deep features. To simplify the model, we adopt the CACL position setting of the ASPP feature fusion layer.

Visualization Result of CACL. The CACL visualization results are shown in the right part of Fig. 5. It can be found that for predictions generated by the same clicks, CACL tends to improve the prediction of SClg-cls in each image, while can maintaining the segmentation performance of other classes with lower semantic challenge degree.

E. Comparisons with the State-of-the-arts

We compare our CM^2 with state-of-the-art interactive semantic segmentation methods on 3 public datasets. It should be noted that ITIS [45] and CA-learning [49] are not specifically designed for the interactive semantic segmentation task. To ensure a fair comparison with them, we adapted their methods to the same base network as CM^2 and conducted individual training on each semantic segmentation dataset, keeping the relevant parameters consistent with our model (refer to "Implementation Details"). Specifically, for ITIS, we adopted the same initial clicking (central clicks) simulation strategy as CM^2 and used the same iterative additional clicking strategy as ITIS during the training process. For CA-learning, we fine-tuned the learning rate during the online learning process on the basis of the already trained interactive semantic segmentation base model (refer to "Implementation Details").

As shown in Tab. II, CM^2 can achieve the preset mIoU with the least number of interactions, which **saves around $\sim$20%, 25% and 30.76% of correction efforts** on LIP (to achieve 85% mIoU), Cityscapes (to achieve 89% mIoU) and PASCAL-Person-Part (to achieve 85% mIoU) datasets, respectively. As shown in Fig. 7, for the same click number, the performance gains can reach about 1.5%, 0.8% and 1.3% mIoU higher than the state-of-the-arts on LIP, Cityscapes and PASCAL-Person-Part datasets, respectively.

Due to the scarcity of existing interactive semantic segmentation methods, we compared our method with powerful interactive object segmentation methods to more comprehensively showcase the efficiency of our mothod. Following the CM [48], we treat each semantic class as an individual object for segmentation (class-agnostic) in testing. Upon completion of segmentation for all the classes in an image, we merge all segmented objects and assign the corresponding semantic labels to achieve the final semantic segmentation result. For a fair comparison, we trained these methods on the same datasets used in this paper and adopt the same initialization clicks located at the center of the semantic class in testing. The results are shown in Fig. 8. Our CM^2 surpasses existing interactive object segmentation methods in interaction efficiency across all three datasets, validating the efficiency of our approach.

At the same time, to investigate the effectiveness of CM^2 on interactive instance-level segmentation. We train our model on the PASCAL VOC 2012 [60] dataset following the CA-learning [49] and construct comparisons on four instance

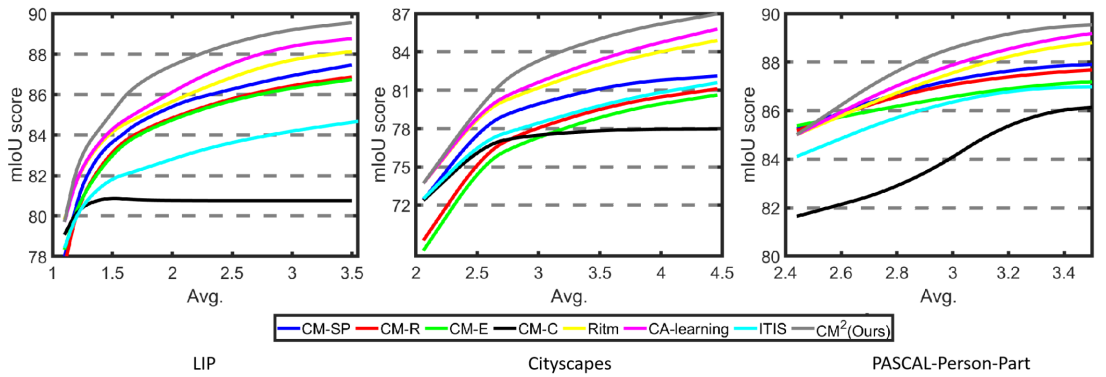


Fig. 7. Quantitative comparison to state-of-the-art interactive semantic segmentation methods on three benchmarks."Avg." represents the average click number on cumulative categories.

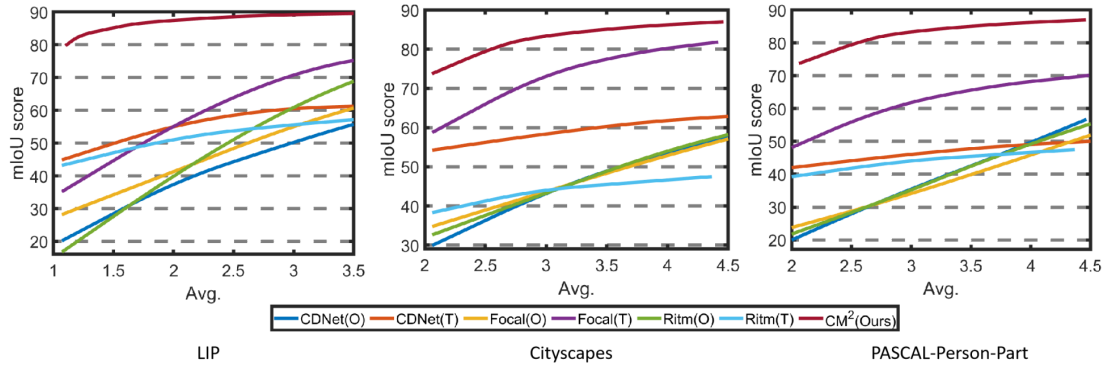


Fig. 8. Quantitative comparison to state-of-the-art interactive object segmentation methods on three benchmarks."Avg." represents the average click number on cumulative categories. "O" represents the original trained model provided by the method. "T" represents the method is retrained on the testing dataset.

TABLE III
QUANTITATIVE COMPARISONS TO STATE-OF-THE-ART METHODS OF THE AVERAGE CLICKING EFFORTS OF ACHIEVING TARGET IOU PERFORMANCE ON INTERACTIVE INSTANCE SEGMENTATION TASK. CM^2 REPRESENT OUR MODEL TRAINED ON PASCAL VOC DATASET, RITM+CM^2 REPRESENT THE RITM [57] MODEL TRAINED ON SBD DATASET ADOPTING HRNET-18 [58] BACKBONE WITH OUR CACL AND CMM MODULE, FOCAL+CM^2 REPRESENT FOCAL [59] MODEL TRAINED ON SBD DATASET ADOPTING SEGFORMERB0-S2 [22] BACKBONE WITH OUR CACL AND CMM MODULE. THE "BK", "VOC", "GC", AND "SBD" REPRESENT THE BERKELEY, PASCAL VOC 2012, GRABCUT, AND SBD DATASET RESPECTIVELY.

	Bk [54] @90% IoU	VOC [60] @85% IoU	GC [61] @90% IoU	SBD [62] 85% IoU
iFCN [55]	8.65	6.88	6.04	-
RIS [46]	6.03	5.12	5.00	6.03
TSLFN [63]	6.49	4.58	3.76	-
CAG [40]	5.60	3.62	3.58	-
BRS [41]	5.08	-	3.60	6.59
CD-Net [42]	3.69	-	2.64	4.37
CA-learning* [49]	4.99	3.47	3.37	4.87
Ritm [57]	4.12	2.94	2.41	3.95
Focal [59]	3.14	3.39	1.90	4.34
CM^2(ours)	4.73	3.25	3.14	4.64
Ritm+CM^2(ours)	3.99	2.31	2.35	3.70
Focal+CM^2(ours)	3.02	3.07	1.82	3.98

segmentation datasets [54], [60]–[62]. To show the universality of our method, in addition to the model with backbone of DeepLab V3+, we have also integrated the CACL and CMM modules into the RITM [57] model that uses HRNet-18 [58] as its backbone and the SBD dataset [62] for training. Furthermore, we have applied these modules to the Focal [59] model with SegformerB0-S2 [22] as its backbone, also trained on the SBD dataset. We report the result in Tab. III, on the basis of different methods, we incorporated the proposed CM^2, and in all cases observed improved performance. which proves that our method can also provide interaction efficiency gains for interactive instance segmentation.

We present a visual comparison of the proposed method with other state-of-the-art methods in Fig. 9. It can be seen that our method achieves superior segmentation results with similar clicking efforts.

F. Other Experimental Results

The Cluster of Correction-misprediction Pairs. We point out that the correction-misprediction pairs are often clustered in one semantic class, *i.e.*, SClg-cls, with higher interaction difficulty. To confirm this, we conduct statistical experiments on the correction click numbers on the LIP and Cityscapes validation set. For each image, we perform 5 / 10 / 15 corrections and sort the semantic classes according to the correction click number they contain. Then we accumulate the correction click number with the same ranking in all images and normalize the statistical correction click numbers. It can be seen from Fig. 10, more than a half of correction clicks are

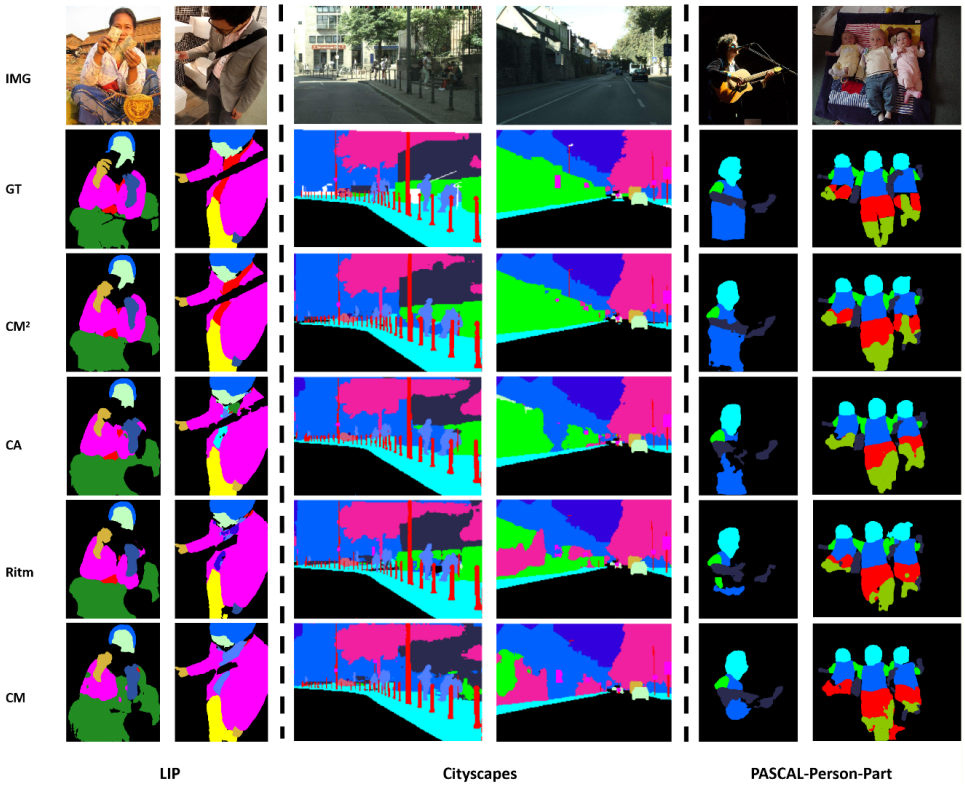


Fig. 9. Comparison with state-of-the-art visualizations. "CA" represents the CA-learning [49] method.

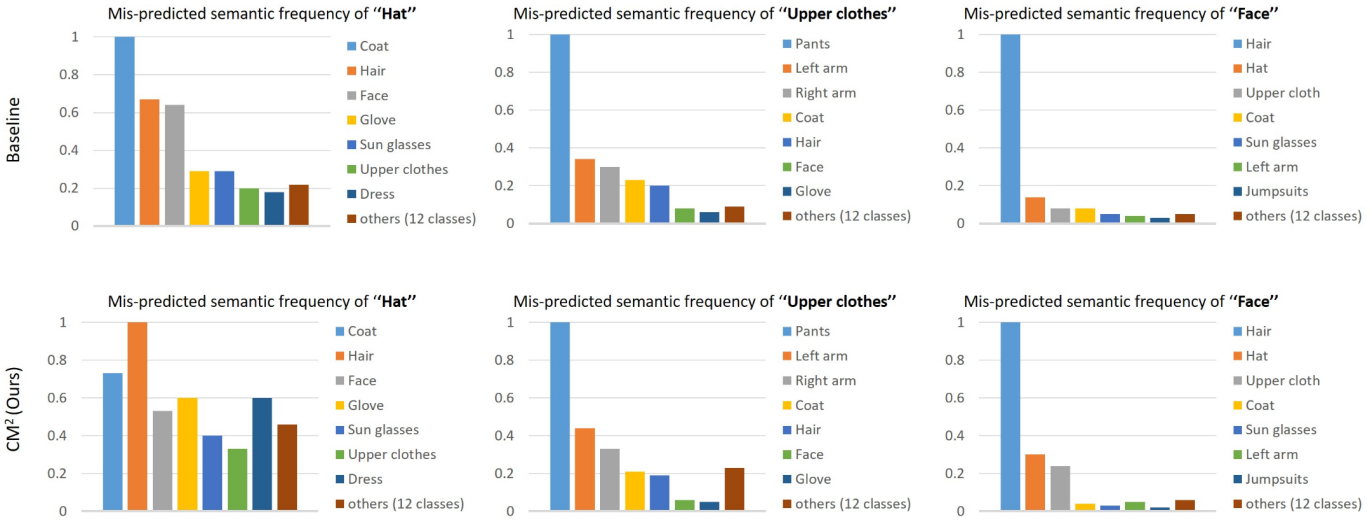


Fig. 11. The frequency of mislabeled classes for semantic classes "Hat", "Upper clothes" and "Face". The corr-times indicates the correction click number added for each category.

located in the same semantic class during the interaction. As corrections increase, the obvious errors have been corrected, the correction clicks tend to locate in narrow error areas, *e.g.*, category adjacent area, and the aggregation of the corrected category distribution is reduced, however, still exist more than a half correction clicks clustered in the same category. This indicates there usually exists an SClg-cls with more correction-misprediction pairs, which supporting our design of CACL.

The Semantic Correlations between Corrections and Mis-predictions. In order to investigate the semantic correlation between user corrections and model predictions, we select 3 semantic classes from the LIP dataset and count the frequency of their missegmented human-part classes through correction clicks, as shown in Fig. 11. For the baseline model, the mispredicted results of semantic classes "Hat", "Upper clothes" and "Face" are frequently repeated in several classes, indicating that the correction-misprediction pairs are semantically correlated, which makes their features regular and supportive for our design of CMM. For our method, it can be found that the frequency of "Hat", "Upper clothes" and "Face" categories being misclassified as specific categories is reduced to different degrees, especially for the "hat" category. This reflects the capability of our method to mitigate the occurrence frequency of the same confusion categories.

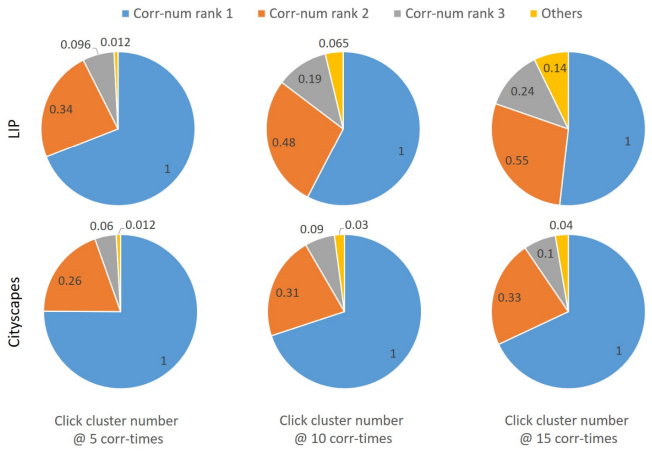


Fig. 10. Correction click number statistics based on the amount of correction-misprediction pairs.

Testing Complexity. The computational complexity and memory usage of our proposed model is similar to the CA-learning. Specifically, 3,636 MB of GPU memory is consumed during the interaction process, and the model inference time for one interaction round is about 0.38 s on one TITAN RTX GPU, the process of generating confusion prototypes using the momentum parameter takes approximately 80 ms. It is worth mentioning that in practical use, the model can make use of the user's correction time for parallel online learning. Therefore, the online learning process has little impact on the annotation effect.

Limitations. Our paper focuses on interactive semantic segmentation. Extant research [48] indicates that interactive semantic segmentation models show promising potential and high interaction efficiency when addressing tasks with a moderate amount of semantic categories, such as human parsing and scene parsing which with relatively fixed categories (around 20). Nonetheless, when the semantic class number is quite large (for instance, in the ADK200 [] dataset), user interaction efficiency may decline due to the increasing difficulty in identifying misprediction regions in practical usage. Simultaneously, the parameter for our CACL and CMM grows with increasing semantic classes. Specifically, the quantity of CACL parameters becomes $(category - number + 1) \times 1280 \times 256$, and the number of parameters in the memory bank stored in the CMM equals $5 \times category - number \times 256$. Therefore, our CM^2 is particularly suited for semantic segmentation tasks involving a moderate quantity of categories, like human parsing and scene parsing.

While our method excels in executing semantic segmentation tasks involving a moderate number of categories, our method is not designed for tackling unseen semantic classes like its counterpart in interactive object segmentation. The robustness and efficiency exhibited in specified tasks, *i.e.*, human parsing and scene parsing, make it notably useful in improving the user experience in corresponding human-machine interaction applications, including motion gaming [64]–[66], and semantic data labeling [33], [48], [57], [59]. Despite its promising performance, the challenge on enabling interactive semantic segmentation models to segment unseen semantic categories remains a meaningful line of inquiry for future studies.

Incorrect clicks provided by users will have a great impact on our model. In the training process, our model is based on the premise that the clicks provided by users are completely correct. Therefore, our method is less robust to inappropriate user clicks. However, in practical applications, users can reduce the frequency of mislabeling by being familiar with the labeling process. In addition, when mislabeling occurs, users can remove the click or add additional clicks to reduce the mislabel impact on the model.

V. Conclusion

We explore the semantic correlation between user corrections and model mispredictions. Based on this, we propose an online learning solution for interactive semantic segmentation, termed as CM^2. It includes a CMM for reducing similar mispredictions and a CACL for enhancing the capability of processing classes that are difficult to segment. Despite its simplicity, extensive experiments show the effectiveness of CM^2 on improving interaction efficiency. Furthermore, our CM^2 can be conveniently used without additional training. These demonstrate that our CM^2 is a simple and effective interactive segmentation solution.

VI. Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant 62072027, and the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).

References

[1] T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, and Y. Zhao, "Devil in the details: Towards accurate single and multiple human parsing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4814–4821.

[2] T. Li, Z. Liang, S. Zhao, J. Gong, and J. Shen, "Self-learning with rectification strategy for human parsing," *In CVPR*, 2020.

[3] X. Zhang, Y. Chen, B. Zhu, J. Wang, and M. Tang, "Part-aware context network for human parsing," *In CVPR*, 2020.

[4] T. Huang, X. U. Yongchao, S. Bai, Y. Wang, and X. Bai, "Feature context learning for human parsing," *China Science*, vol. 062, no. 012, pp. P.2–15, 2019.

[5] X. Liang, K. Gong, X. Shen, and L. Lin, "Look into person:joint body parsing and pose estimation network and a new benchmark." *In TPAMI*, vol. 41, no. 4, pp. 871–885, 2018.

[6] D. Zeng, Y. Huang, Q. Bao, J. Zhang, C. Su, and W. Liu, "Neural architecture search for joint human parsing and pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 385–11 394.

[7] V. R. Kumar, M. Klingner, S. Yogamani, S. Milz, T. Fingscheidt, and P. Mader, "Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 61–71.

[8] A. Sagar and R. Soundrapandiyan, "Semantic segmentation with multi scale spatial attention for self driving cars," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2650–2656.

[9] C. Lin, D. Tian, X. Duan, J. Zhou, D. Zhao, and D. Cao, "3d-dfm: Anchor-free multimodal 3-d object detection with dynamic fusion module for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[10] J. Wu, Z. Huang, W. Huang, and C. Lv, "Prioritized experience-based reinforcement learning with human guidance for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[11] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[12] Y. Ding, F. Tan, Z. Qin, M. Cao, K.-K. R. Choo, and Z. Qin, "Deepkeygen: a deep learning-based stream cipher generator for medical image encryption and decryption," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[13] M. Rezaei, H. Yang, and C. Meinel, "Recurrent generative adversarial network for learning imbalanced medical image semantic segmentation," *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15 329–15 348, 2020.

[14] S. Gerhard, J. Funke, J. Martel, A. Cardona, and R. Fetter, "Segmented anisotropic sstem dataset of neural tissue," *figshare*, pp. 0–0, 2013.

[15] A. Suinesiaputra, B. R. Cowan, A. O. Al-Agamy, M. A. Elattar, N. Ayache, A. S. Fahmy, A. M. Khalifa, P. Medrano-Gracia, M.-P. Jolly, A. H. Kadish *et al.*, "A collaborative resource to build consensus for automated left ventricular segmentation of cardiac mr images," *Medical image analysis*, vol. 18, no. 1, pp. 50–62, 2014.

[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask rcnn." *In ICCV*, 2017.

[17] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille., "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *In TPAMI*, vol. 40, no. 4, pp. 834–848, 2018.

[18] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[19] Y. Yuan and J. Wang, "Ocnet: Object context network for scene parsing." *In arXiv preprint arXiv:1809.00916*, 2018.

[20] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7262–7272.

[21] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[22] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[23] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7303–7313.

[24] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, "Rethinking bisenet for real-time semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9716–9725.

[25] L. Zhu, D. Ji, S. Zhu, W. Gan, W. Wu, and J. Yan, "Learning statistical texture for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 537–12 546.

[26] A. Ouahabi and A. Taleb-Ahmed, "Deep learning for real-time semantic segmentation: Application in ultrasound imaging," *Pattern Recognition Letters*, vol. 144, pp. 27–34, 2021.

[27] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1568–1576.

[28] Z. Zhang, G. Gao, Z. Fang, J. Jiao, and Y. Wei, "Mining unseen classes via regional objectness: A simple baseline for incremental segmentation," *Advances in neural information processing systems*, vol. 35, pp. 24 340–24 353, 2022.

[29] Z. Zhang, G. Gao, J. Jiao, C. H. Liu, and Y. Wei, "Coinseg: Contrast inter-and intra-class representations for incremental segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 843–853.

[30] K. K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation." *In CVPR*, 2018.

[31] Z. Wang, D. Acuna, H. Ling, A. Kar, and S. Fidler, "Object instance annotation with deep extreme level set evolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7500–7508.

[32] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang., "Deep grabcut for object selection." *In BMVC*, 2017.

[33] S. Zhang, J. H. Liew, Y. Wei, S. Wei, and Y. Zhao, "Interactive object segmentation with inside-outside guidance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 234–12 244.

[34] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation." *In ICCV*, 2017.

[35] H. Le, L. Mai, B. Price, S. Cohen, H. Jin, and F. Liu, "Interactive boundary prediction for object selection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 18–33.

[36] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++." *In CVPR*, 2018.

[37] M. Andriluka, J. R. Uijlings, and V. Ferrari, "Fluid annotation: a human-machine collaboration interface for full image annotation," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1957–1966.

[38] E. Agustsson, J. R. R. Uijlings, and V. Ferrari, "Interactive full image segmentation by considering all regions jointly." *In CVPR*, 2019.

[39] D.-J. Chen, J.-T. Chien, H.-T. Chen, and L.-W. Chang, "Tap and shoot segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[40] S. Majumder and A. Yao, "Content-aware multi-level guidance for interactive instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 602–11 611.

[41] W. D. Jang and C. S. Kim, "Interactive image segmentation via backpropagating refinement scheme." *In CVPR*, 2019.

[42] X. Chen, Z. Zhao, F. Yu, Y. Zhang, and M. Duan, "Conditional diffusion for interactive segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7345–7354.

[43] K. Sofiiuk, I. Petrov, O. Barinova, and A. Konushin, "f-brs: Rethinking backpropagating refinement for interactive segmentation," *In CVPR*, 2020.

[44] Z. Lin, Z. Zhang, L.-Z. Chen, M.-M. Cheng, and S.-P. Lu, "Interactive image segmentation with first click attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 339–13 348.

[45] S. Mahadevan, P. Voigtlaender, and B. Leibe, "Iteratively trained interactive segmentation," in *British Machine Vision Conference (BMVC)*, 2018, pp. 1–12.

[46] J. H. Liew, Y. Wei, W. Xiong, S. H. Ong, and J. Feng, "Regional interactive image segmentation networks," *In ICCV*, 2017.

[47] Z. Li, Q. Chen, and V. Koltun, "Interactive image segmentation with latent diversity." *In CVPR*, 2018.

[48] Y. Gao, L. Liang, C. Lang, S. Feng, Y. Li, and Y. Wei, "Clicking matters: Towards interactive human parsing," *IEEE Transactions on Multimedia*, 2022.

[49] T. Kontogianni, M. Gygli, J. Uijlings, and V. Ferrari, "Continuous adaptation for interactive object segmentation by learning from corrections," in *European Conference on Computer Vision*. Springer, 2020, pp. 579–596.

[50] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[51] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan, "Human parsing with contextualized convolutional neural network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1386–1394.

[52] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[53] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. 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*, 2014, pp. 1971–1978.

[54] K. McGuinness and N. E. Oconnor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[55] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection." *In CVPR*, 2016.

[56] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[57] K. Sofiiuk, I. A. Petrov, and A. Konushin, "Reviving iterative training with mask guidance for interactive segmentation," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 3141–3145.

[58] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[59] X. Chen, Z. Zhao, Y. Zhang, M. Duan, D. Qi, and H. Zhao, "Focalclick: Towards practical interactive image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1300–1309.

[60] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep*, vol. 2007, pp. 1–45, 2012.

[61] C. Rother, V. Kolmogorov, and A. Blake, "" grabcut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[62] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *2011 international conference on computer vision*. IEEE, 2011, pp. 991–998.

[63] Y. Hu, A. Soltoggio, R. Lock, and S. Carter, "A fully convolutional two-stream fusion network for interactive image segmentation," *Neural Networks*, vol. 109, pp. 31–42, 2019.

[64] Y. Liu, Y. Song, and R. Tamura, "Hedonic and utilitarian motivations of home motion-sensing game play behavior in china: An empirical study," *International journal of environmental research and public health*, vol. 17, no. 23, p. 8794, 2020.

[65] A. Singh and T. S. Parihar, "Prospects of customization in educational pedagogy through motion sensing interactive educational games in india," *Specialusis Ugdymas*, vol. 1, no. 43, pp. 6855–6862, 2022.

[66] H.-J. Hu, "A study of interactive design games to enhance the fun of muscle strength training for older adults," in *International Conference on Human-Computer Interaction*. Springer, 2023, pp. 420–434.

Yutong Gao received the PhD degree in the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. From 2021 to 2022, he visited Institute for Infocomm Research (I2R), Agency for Science, Technology & Research (A*STAR), Singapore, as a Visiting Researcher. He received the MA. degree from National Taipei University of Technology, Taipei, China, in 2017. He received the B.S. degree from Northeastern University, Liaoning, China. He is now a lecturer at the School of Information Engineering of Minzu University of China. His current research interests include computer vision and machine learning.

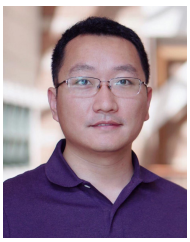
Congyan Lang received the Ph.D. degree from the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China, in 2006. She was a Visiting Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from 2010 to 2011. From 2014 to 2015, she visited the Department of Computer Science, University of Rochester, Rochester, NY, USA, as a Visiting Researcher. She is currently a Professor with the School of Computer and Information Technology, Beijing Jiaotong University. Her current research interests include multimedia information retrieval and analysis, machine learning, and computer vision.

Fayao Liu Fayao Liu is a research scientist at Institute for Infocomm Research, A*STAR, Singapore. She received her PhD in computer science from University of Adelaide, Australia in Dec. 2015. She mainly works on machine learning and computer vision problems, with particular interests in self-supervised learning, few-shot learning and generative models.

Chuan-Sheng Foo received his BS, MS and PhD from Stanford University. He currently leads a research group at the Institute for Infocomm Research, A*STAR, Singapore, which focuses on developing data-efficient deep learning algorithms that can learn from less labeled data.

Yuanzhouhan Cao Yuanzhouhan Cao is currently a lecturer at the School of Computer Science and Information Technology, Beijing Jiaotong University. He has obtained Master of Engineering and PhD of Computer Science from Sichuan University, China, and The University of Adelaide, Australia, in 2013 and 2017 respectively. He was a research scientist at Idiap Research Institute, Switzerland from 2017 to 2018, and a lecturer in the University of Adelaide from 2019 to 2020. His research interests are computer vision, machine learning and deep learning.

Lijuan Sun Lijuan Sun received the Ph.D. degree in the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, P.R. China, in 2021. She is currently an associate researcher in the Intellectual Property Information Service Center, Beihang University. She is currently a key member of Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education. Her research interests include machine learning and computer vision.



Yunchao Wei Yunchao Wei is currently a full professor at Beijing Jiaotong University. Before join BJTU, he conducted research at the National University of Singapore, the University of Illinois at Urbana-Champaign and the University of Technology Sydney. He was selected as MIT TR35 China by MIT Technology Review in 2021, Global Outstanding Chinese Young Scholar by Baidu in 2021, and was named as one of the five top early-career researchers in Engineering and Computer Sciences in Australia by The Australian in 2020. He has received numerous awards, including the First Prize of the Ministry of Education Natural Science Award (2022), First Prize of the Science and Technology Award from the China Society of Image and Graphics (2019), Young Researcher Award from the Australian Research Council (2019), IBM C3SR Best Research Award (2019). He received many competition prizes from CVPR/ICCV/ECCV such as the Winner prizes of ILSVRC 2014, LIP 2018/2019, Youtube VOS 2021, Runner-up Prizes of ILSVRC 2017, DAVIS 2020, etc. He has published over 100 papers in top journals/conferences such as TPAMI and CVPR, with around 20,000 citations on Google Scholar. His current research focuses on visual perception with imperfect data, multimodal data analysis, generative artificial intelligence, and more.