Continual Learning for Image Segmentation With Dynamic Query

Weijia Wu, Yuzhong Zhao, *Member, IEEE*, Zhuang Li, Lianlei Shan, Hong Zhou, and Mike Zheng Shou, *Member, IEEE*

***Abstract*— Image segmentation based on continual learning exhibits a critical drop of performance, mainly due to catastrophic forgetting and background shift, as they are required to incorporate new classes continually. In this paper, we propose a simple, yet effective Continual Image Segmentation method with incremental Dynamic Query (CISDQ), which decouples the representation learning of both old and new knowledge with lightweight query embedding. CISDQ mainly includes three contributions: 1) We define *dynamic queries* with adaptive background class to exploit past knowledge and learn future classes naturally. 2) CISDQ proposes a class/instance-aware Query Guided Knowledge Distillation strategy to overcome catastrophic forgetting by capturing the inter-class diversity and intra-class identity. 3) Apart from semantic segmentation, CISDQ introduce the continual learning for *instance segmentation* in which instance-wise labeling and supervision are considered. Extensive experiments on three datasets for two tasks (*i.e.,* continual semantic and instance segmentation are conducted to demonstrate that CISDQ achieves the state-of-the-art performance, specifically, obtaining 4.4% and 2.9% mIoU improvements for the ADE 100-10 (6 steps) setting and ADE 100-5 (11 steps) setting.**

***Index Terms*— Image segmentation, continual learning, dynamic query, dense matching strategy, transformer.**

I. Introduction

IMAGE segmentation, including semantic segmentation and instance segmentation, is a fundamental task in computer vision. In recent years, data-driven segmentation networks [1], [2] have made extraordinary progress with fully-supervised learning of fixed data, where all classes are fixed and known beforehand and learned at once. However, in a real-world system, it is preferable that one model can dynamically update its knowledge and extend to segment new classes without retraining from scratch. To reach the setup, previous works [3], [4], [5] propose to introduce the class incremental learning for semantic segmentation, named continual semantic segmentation (CSS). Different from these works that focus on semantic segmentation, in this paper, we try to establish a generic continual image segmentation task, which includes semantic segmentation and instance segmentation.

Continual learning-based image segmentation (or continual image segmentation, CIS) is a task that requires one model incrementally learn and segment newly arriving class objects while not catastrophically forgetting the past learned classes (old classes). Similar to continual semantic segmentation, continual image segmentation also faces two main challenges. The first one is the **catastrophic forgetting** [6], [7], where the model quickly fits the new data distribution and loses the old discriminative representation. During continual learning, the new training data and class annotation is only used, and the old classes are not available usually be treated as background. The network usually tends to catastrophically and abruptly forget previously learned knowledge (old classes) when learning new information (new classes). Existing continual semantic segmentation methods [3], [5] typically adopt inefficient and explicit multiple strategies to tackle the problem. For instance, PLOP [3] naively adopts a spatial distillation loss without fine-grained relationship learning of intra-class, inter-class, each pixel. SSUL [5] adopts three strategies, *i.e.,* pseudo-labeling, exemplar memory, and model freezing to cover the challenge, but pseudo-labeling usually shows extremely instability and exemplar memory requires the extra cost memory of the network.

The second challenge, inherited from continual semantic segmentation, is the **background shift** [3], [5]. Different from the fixed and certain background pixels of traditional image segmentation, the background pixel for continual semantic segmentation belongs to *three categories*: potential future object classes, past object classes, and the true background. For instance, if an image contains three object classes, *i.e.,* `sofa`, `person`, `dog`, where only class `sofa` mask annotation is available and class `person` and `dog` belong to the past classes without annotation. If the model naively treats all unknown background pixels to the true background pixel class, the old and future knowledge will be further damaged and forgotten. Some works [3], [5] try to solve background shift

Manuscript received 27 July 2023; revised 29 October 2023; accepted 26 November 2023. Date of publication 30 November 2023; date of current version 6 June 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3602601, in part by the Key Research and Development Program of Zhejiang Province of China under Grant 2021C02037. The work of Mike Zheng Shou was supported in part by Mike Zheng Shou's Start-Up Grant from NUS; and in part by the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 under Grant FY2022. This article was recommended by Associate Editor H. Zhang. *(Weijia Wu and Yuzhong Zhao contributed equally to this work.) (Corresponding author: Hong Zhou.)*

Weijia Wu and Hong Zhou are with the College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou 310058, China (e-mail: weijiawu@zju.edu.cn; zhouh@mail.bme.zju.edu.cn).

Yuzhong Zhao and Lianlei Shan are with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China.

Zhuang Li is with Kuaishou Technology, Beijing 100000, China.

Mike Zheng Shou is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3337884.

Digital Object Identifier 10.1109/TCSVT.2023.3337884

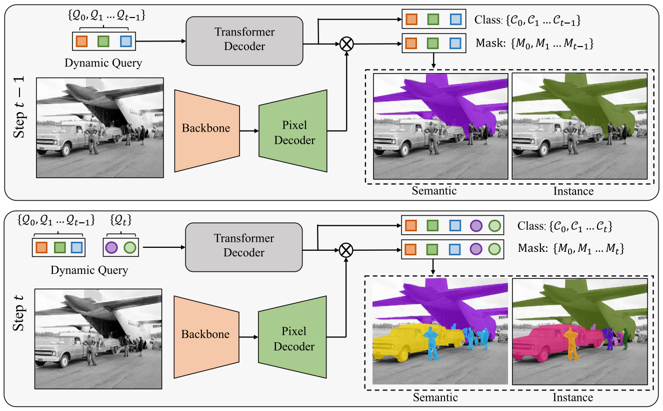

$\{\mathcal{Q}_0, \mathcal{Q}_1 \ldots \mathcal{Q}_{t-1}\}$
Dynamic Query
Transformer Decoder
Class: $\{\mathcal{C}_0, \mathcal{C}_1 \ldots \mathcal{C}_{t-1}\}$
Mask: $\{M_0, M_1 \ldots M_{t-1}\}$
Step $t-1$
Backbone
Pixel Decoder
Semantic
Instance
$\{\mathcal{Q}_0, \mathcal{Q}_1 \ldots \mathcal{Q}_{t-1}\}$ $\{\mathcal{Q}_t\}$
Dynamic Query
Transformer Decoder
Class: $\{\mathcal{C}_0, \mathcal{C}_1 \ldots \mathcal{C}_t\}$
Mask: $\{M_0, M_1 \ldots M_t\}$
Step t
Backbone
Pixel Decoder
Semantic
Instance

Fig. 1. **Illustration of CISDQ**. Dynamic query decouples the representation learning of old and new knowledge with dynamically increasing lightweight queries. Q_t, C_t and M_t refers to the Query group, Classes, and Masks in t-th step, respectively.

with pseudo-labeling from the old model or other off-the-shelf saliency-map detectors. However, the predicted pseudo-labeling from the model usually present instability and worse performance, and it is impossible to predict all potential objects for future classes.

In this paper, we propose a simple, yet efficient **C**ontinual **I**mage **S**egmentation method with incremental **D**ynamic **Q**uery (**CISDQ**), which decouples the representation learning of both old and new knowledge with class/instance-aware lightweight query embedding. Inspired by query-based modeling for vision tasks [2], [8], [9], [10], [11], we propose a new concept *dynamic query*, enabling the model dynamically update its knowledge and extend to segment new classes via dynamically increasing query embedding of new classes or instances. As shown in Fig. 1, we adopt one query embedding to represent one class or instance and dynamically increase the number of query embeddings to extend new classes without retraining from scratch. **CISDQ** main includes two advantages: 1) Different queries are used to represent different classes at each step to handle *catastrophic forgetting* problem. At step t, queries $\{Q_0, Q_1 \ldots Q_{t-1}\}$ will be used to retain the past knowledge of old classes (*i.e.,* $\{\mathcal{C}_0, \mathcal{C}_1 \ldots \mathcal{C}_{t-1}\}$), newly added queries $\{Q_t\}$ are used to learn new appeared knowledge (classes) $\{\mathcal{C}_t\}$. 2) *Adaptive* background concept for each query of each step is proposed to solve *background shift* problem. Different background definitions $\{\mathcal{B}_0, \mathcal{B}_1 \ldots \mathcal{B}_{t-1}\}$ for different queries $\{Q_0, Q_1 \ldots Q_{t-1}\}$ are used to cover different background representation at each step. CNN-based multi-classification per-pixel in one feature map [1] is transformed to a set of binary-classification in multi-maps [2], thus there are t different definitions of background for t predicted masks each associated with a single category. For instance, at step $t-1$, if only class `sofa` mask annotation is available, `person`, `dog` pixel region as the *unknown* region is defined as the background $\mathcal{B}_{t-1}$. But at step t, `person` as newly added class is defined as the positive sample, but `sofa` mask is not available, belonging to the background class $\mathcal{B}_t$. Thus the adaptive background does not give confusing supervision, while there is no background shift.

Besides, we propose a query-guided knowledge distillation (Query G-KD) to further alleviate catastrophic forgetting. Different from previous knowledge distillation of semantic segmentation, Query G-KD can achieve more precise class/instance-aware knowledge distillation via capturing the inter-class diversity and intra-class identity, while each query is responsible for one class. Similarly, the knowledge of inter-instance diversity and intra-instance identity can be retained with Query G-KD for continual instance segmentation tasks. To summarize, our contributions are four-folds:

- We propose a simple, yet efficient unified **C**ontinual **I**mage **S**egmentation method with incremental **D**ynamic **Q**uery, namely **CISDQ**, including a *adaptive* background concept for each class/instance, which decouples the representation learning of old and new knowledge with dynamically increasing lightweight query embedding.
- **CISDQ** introduces a Query Guided Knowledge Distillation (Query G-KD) strategy to overcome catastrophic forgetting via capturing the inter-class diversity and intra-class identity.
- **CISDQ** introduces continual learning for *instance segmentation* in which instance-wise labeling and supervision are considered. We also solve semantic and instance segmentation in a unified framework.
- Experiments are conducted on *continual semantic segmentation* and *continual instance segmentation* on three datasets, respectively. And **CISDQ** achieves state-of-the-art performance with up to 10% improvements than previous works.

II. Related Work

A. Image Segmentation

Image Segmentation mainly includes two tasks: semantic and instance segmentation tasks. For semantic segmentation, Some early works attempted [12], [13], [14], [15], [16] to explore the use of certain image priors, such as Gaussian Mixture Modeling, to segment images or videos. Fully Convolutional Networks (FCN) [17] is the first deep learning based work to perform pixel-to-pixel semantic classification in an end-to-end manner. After that, researchers focused on different aspects for improving semantic segmentation, *e.g.,* contextual relationships [18], [19], [20], [21], Spatial pyramid pooling [22], [23], [24], and image pyramid [25], [26]. In recent years, several works [27], [28] try to adopt transformer architectures for semantic segmentation. Segformer [27] design a novel hierarchically structured Transformer encoder to output multi-scale features. For instance segmentation [29], Mask R-CNN [30], as the representation, extend Faster R-CNN [31] by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. DETR [32] proposed to segment instances with mask attention, which is more natural. Maskformer [28] proposed mask classification modeling, which solves semantic- and instance-level segmentation tasks in a unified manner. GenPromp [33], DiffuMask [34] and DatasetDM [35] proposed to use diffusion model for enhancing semantic segmentation tasks, supporting open-set segmentation. In this paper, we try to utilize the advantage of a transformer for continual image segmentation.

B. Class Continual Learning

Class continual learning method [36], [37], [38], [39], [40], [41], [42] mainly focuses on the classification task, and alleviates catastrophic forgetting [43], which is caused by the domain distribution change from the training data. To solve this problem, most works [44], [45], [46] try to maintain the performance of old classes with knowledge distillation [47], [48], [49], and adversarial training [50], [51]. In recent years, inspired by the success of transformer architecture [32], [52] in computer vision, some works try to solve class incremental learning with transformer [53], [54], [55], [56]. MEta-ATtention [53] proposed to use a pre-trained ViT to new tasks without sacrificing performance on already learned tasks. DyTox [55] designs a shared transformer encoder and decoder to cover incremental learning with a dynamic expansion of special tokens. TwF [56] proposed a hybrid approach building upon a fixed pre-trained sibling network, which continuously propagates the knowledge inherent from the previous task through a layer-wise loss term. However, no one tries to adopt transformer to solve continual semantic and instance segmentation.

C. Continual Image Segmentation

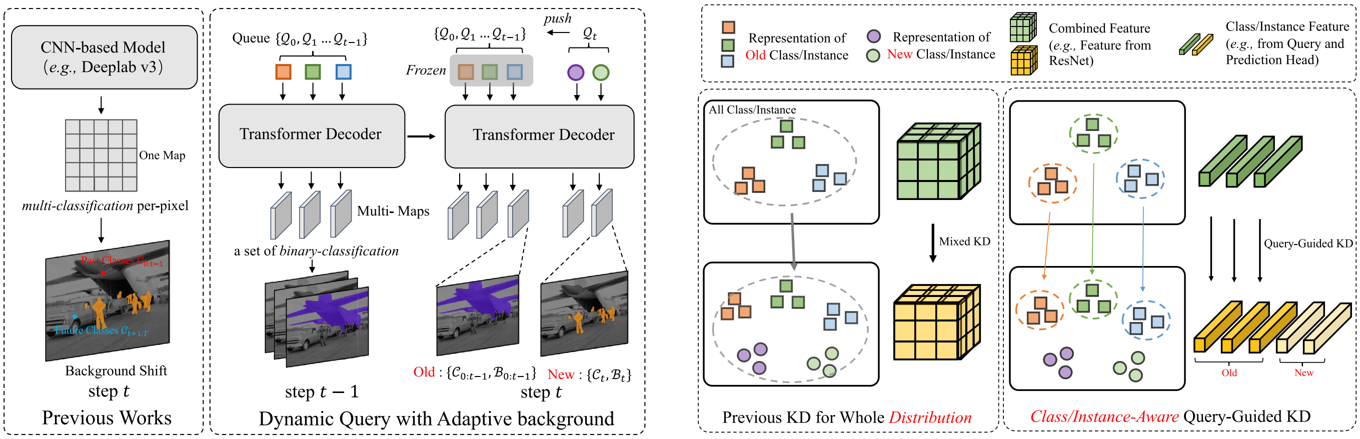
1) Semantic Segmentation: Except for catastrophic forgetting, continual semantic segmentation also faces another challenge, *i.e.,* background shift [3], [5]. Most existing methods solve the two challenges with rehearsal-based [57], [58], pesudo label [3], [59], [60], [61], and knowledge distillation [3], [62]. MiB [62] propose a new objective function and introduce a specific classifier initialization strategy to solve the background shift. PLOP [3] proposed a multi-scale pooling distillation and entropy-based pseudo-labelling of the background to deal with the two problems. RCIL [4] proposed a structural re-parameterization to decouple the representation learning of both old and new knowledge for solving catastrophic forgetting. SSUL [5] solves background shift and catastrophic forgetting with three strategies, *i.e.,* unknown classes in background class, freeze backbone, and exemplar memory. To avoid forgetting old knowledge, MicroSeg [63] first splits the given image into hundreds of segment proposals with a proposal generator. Those segment proposals with strong objectness from the background are then clustered and assigned newly-defined labels during the optimization. Based on Segformer [27], SATS [64] design a knowledge distillation for transformer-based semantic segmentation. However, the above method still can not achieve preferable performance, while they can not solve the two challenges entirely. Different from previous methods, We propose an incremental dynamic query and query-guided knowledge distillation to decouple the representation learning of old and new knowledge.

2) Instance Segmentation: Despite enormous progress in the continual learning and instance segmentation tasks, almost no work try to solve continual problem in instance segmentation, while continual instance segmentation is also important And there exists a related task, *i.e.,* incremental few-shot instance segmentation. iMTFA [65] design the first approach for incremental few-shot instance segmentation, which match these class embeddings at the RoI-level using cosine similarity. iFS-RCNN [66] leverages Bayesian learning to address a paucity of training examples of new classes. But the setting of incremental few-shot instance segmentation is different with continual instance segmentation in two parts: 1). Continual instance segmentation requires model to learn feature of new class during *multi-steps*, while incremental few-shot instance segmentation only includes two-steps. 2). Continual instance segmentation provides abundant data of new classes, while incremental few-shot instance segmentation only training on a few data. Besides, different from the above two methods, the proposed dynamic query and query-guided knowledge distillation can decouple the representation learning of old and new knowledge, which can deal with catastrophic forgetting and background shift challenges effectively.

III. Notations and Problem Setting

Similar to previous works [3], [5], for the continual learning scenario, each training on the newly added classes and dataset as a *step*, where existing T steps. For t-th step, given a base model $f_{\boldsymbol{\theta}}^{t-1}$ with parameter θ_{t-1} trained on $\{\mathcal{D}_0, \mathcal{D}_1 \dots \mathcal{D}_{t-1}\}$ with $\{\mathcal{C}_0, \mathcal{C}_1 \dots \mathcal{C}_{t-1}\}$ classes, the model is expected to segment $\sum_{i=0}^{t} \mathcal{C}_i$ classes after training on the newly added dataset $\mathcal{D}_t$ with extra $\mathcal{C}_t$ new classes, where the training data of old classes are not accessible. The network usually quickly fits the new data distribution with extra $\mathcal{C}_t$ new classes, and cause serious performance drop for old classes. The challenge in continual image segmentation is named *catastrophic forgetting*. In this paper, existing T different query groups $\{Q_0, Q_1 \dots Q_T\}$ are responsible for $\{\mathcal{C}_0, \mathcal{C}_1 \dots \mathcal{C}_T\}$ classes at each step. Therefore, we can decouple the representation learning of old and new knowledge for step t from two aspects: 1) Query groups $\{Q_0, Q_1 \dots Q_{t-1}\}$ will be frozen to retain the past knowledge of old classes (*i.e.,* $\sum_{i=0}^{t-1} \mathcal{C}_i$), newly added queries $\{Q_t\}$ are used to learn new appeared knowledge (classes) $\{\mathcal{C}_t\}$; 2) Knowledge distillation [3], [4], [49] is a commonly used technique to overcome the challenge of catastrophic forgetting, where it typically involves directly distilling and aligning the features between t-th step and $(t-1)$-th to avoid a serious performance drop for old classes when learning new classes. Different from previous knowledge distillation, query-guided knowledge distillation is more precise to transfer the knowledge from $f_{\boldsymbol{\theta}}^{t-1}$ to $f_{\boldsymbol{\theta}}^{t}$ at each class/instance level with query groups $\sum_{i=0}^{t-1} Q_i$, where the inter-class/instance diversity and intra-class/instance identity can be captured, as shown in Fig. 2b. Specifically, existing knowledge distillation methods often directly extract features from the backbone and decoder for distillation. However, features at $t-1$ step only contain characteristics of old classes, when features at step $t-1$ contain characteristics of both old and new classes. Forcing distillation on these two features can undermine the learning of new class features because features at step $t-1$ do not encompass the characteristics of new classes.

For t-th step, we use $\mathcal{D}_t = \{\boldsymbol{x}_t, \boldsymbol{y}_t\}$ to denote the current training set, where $\boldsymbol{x}_t \in \mathcal{X}$ denotes the input image, and the $\boldsymbol{y}_t \in \mathcal{Y}_t$ denotes the corresponding ground-truth (GT) *pixel* labels. The label space $\mathcal{Y}_t = \mathcal{B}_t \cup \mathcal{C}_t$ consists of the current


CNN-based Model
(e.g., Deeplab v3)
One Map
multi-classification per-pixel
Past Classes $\mathcal{C}_{0:t-1}$
Future Classes $\mathcal{C}_{t+1:T}$
Background Shift
step t
Previous Works
Queue $\{Q_0, Q_1 ... Q_{t-1}\}$
Transformer Decoder
Multi- Maps
a set of binary-classification
step $t-1$
$\{Q_0, Q_1 ... Q_{t-1}\}$
push
Q_t
Frozen
Transformer Decoder
Old : $\{\mathcal{C}_{0:t-1}, \mathcal{B}_{0:t-1}\}$
New : $\{\mathcal{C}_t, \mathcal{B}_t\}$
step t
Dynamic Query with Adaptive background
Representation of Old Class/Instance
Representation of New Class/Instance
Combined Feature (e.g., Feature from ResNet)
Class/Instance Feature (e.g., from Query and Prediction Head)
All Class/Instance
Mixed KD
Previous KD for Whole Distribution
Query-Guided KD
Old
New
Class/Instance-Aware Query-Guided KD

(a) Dynamic Query for Continual Image Segmentation.

(b) Class/Instance-Aware Query-Guided KD.

Fig. 2. **Illustration for Dynamic Query and Query-Guided Knowledge Distillation.** (a) Incremental Dynamic Query decouples the representation learning of old and new knowledge with dynamically increasing lightlweight query embedding; (b) Compared with previous knowledge distillation [3], [4], [5] over the whole feature distribution, the proposed Query-Guided KD develop class/instance level distillation to overcome catastrophic forgetting challenge.

newly added classes $\mathcal{C}_t$ and the dummy background class $\mathcal{B}_t$. The $\mathcal{B}_t$ label may includes *past* classes $\mathcal{C}_{0:t-1}$, the *future* classes $\mathcal{C}_{t+1:T}$, or the true background pixels $\widetilde{\mathcal{B}}_t$. Therefore, the dummy background class will cause a serious performance drop, while the potential foreground classes ($\mathcal{C}_{0:t-1}$, $\mathcal{C}_{t+1:T}$) is uncertain or misleading. The challenge in continual image segmentation is named *background shift*. All previous works try to predict the potential foreground classes, *i.e.,* $\mathcal{C}_{0:t-1}$, $\mathcal{C}_{t+1:T}$ with the pseudo label from old model [3], [62] or other off-the-shelf saliency-map detector [5]. But the predicted pseudo-labeling from the model usually presents instability and worse performance, and it is impossible to predict all potential objects for future classes. In our work, we introduce a new concept, "*Adaptive*" background class, set $\mathcal{B}_t = \{c_t\} \cup \widetilde{\mathcal{B}}_t$, where $\{c_t\}$ refer to the potential foreground classes and is considered as background for training at t-th step. c_t is a variable and different for each class in a different step. For instance, at step $t-1$, if only class `sofa` mask annotation is available, `person`, `dog` pixel region as the *unknown* regions belong to $\{c_{t-1}\}$. But at step t, `person` as newly added class is defined as the positive sample, but `sofa` mask is not available, belonging to the 'background' class $\{c_t\}$. The adaptive background concept can completely *solve* the negative impact from background shift, while we do not need to give a certain definition for what is real background in each step.

IV. Approach

A. Dynamic Query

Fig. 2a presents the whole framework for incremental Dynamic Query, which describes how to decouple the retaining of old knowledge and learning of new knowledge. All previous CNN-based works [3], [5] all adopt Deeplab v3 [1] as the base framework, which views segmentation task as the multi-classification per-pixel problem. It is difficult to learn new classes and keep discrimination for old classes simultaneously for such a framework, where all representations are stored in the same embedding space. Different from these works, we adopt query-based mask classification architecture, Mask2former [2], which uses a set of C-dimensional feature vectors (*i.e.,* "query") to predict a set of binary masks each associated with a single category. Therefore, different representations from different categories or instances can be decoupled into different feature embedding with a different query.

1) Incremental Dynamic Query Queue: To enable continual learning in such architecture, we modify the fixed set of learned queries of Transformer decoder [2] to *dynamic* incremental increasing sets. For the standard setting, given a input image I and the base model $f_{\boldsymbol{\theta}}$ with parameter θ, the predicted classes is denoted as $\{\mathcal{C}, \mathcal{B}\} = f_{\boldsymbol{\theta}}(Q|I)$, where Q is a fixed set of queries, *i.e.,* N C-dimensional learnable feature vectors. $\mathcal{C}$ and $\mathcal{B}$ are the predicted foreground classes and background class.

The fixed set of queries is transformed to a dynamic incremental increasing query queue of C-dimensional vectors. Specially, in step 0, Q_0 include n_0^q C-dimensional vectors, representing the base class set $\mathcal{C}_0$ with n_0^c classes, where $n_0^q > n_0^c$. And we model the current background class as $\mathcal{B}_0 = \{c_0\} \cup \widetilde{\mathcal{B}}_0$, where $\widetilde{\mathcal{B}}_0$ is the true background pixel, and c_0 is the potential foreground pixel defined as 'background', *e.g.,* the region of *past* class `person`. For the subsequent learning steps, the model is supposed to segment newly added classes $\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3 \ldots\}$. Corresponding, newly query sets $\{Q_1, Q_2, Q_3 \ldots\}$ will be added to guide the representation learning of newly added classes. In the training phase of step t, query embedding sets $Q_{0:t-1}$ will be frozen to avoid forgetting old knowledge. And Q_t is used to guide the network to learn new knowledge of newly added classes. Finally, we model the continual learning at t-th step as:

$$\{\mathcal{C}_{0:t}, \mathcal{B}_{0:t}\} = f_{\boldsymbol{\theta}}(Q_{0:t-1}, Q_t|I)\,. \tag{1}$$

Actually, only $\{\mathcal{C}_t, \mathcal{B}_t\}$ are supervised by manual annotation, while other annotations of $\{\mathcal{C}_{0:t-1}, \mathcal{B}_{0:t-1}\}$ are not available. Class/instance-aware query guided knowledge distillation (Section IV-B) is used to further retain the old knowledge of past classes.

2) Adaptive Background for Different Query Set: Section IV-A.1 have provided basic information concerning the adaptive backgrounds $\{\mathcal{B}_0, \mathcal{B}_1 \dots \mathcal{B}_T\}$ for different step. Actually, as shown in Fig. 2a, different from foreground classes $\mathcal{C}_{0:T}$, we do not use query to guide predict background mask, where a pixel belongs to background region $\mathcal{B}_t$ if it is not in foreground region $\mathcal{C}_t$ at step t. During continual learning steps, the definition of background will change adaptively with the changing of foreground classes. Therefore, the background shift problem can be solved naturally, while it is unnecessary to give a certain definition that which region is the true background, and no confused supervision.

3) Independent Matching for Each Query Set: Independent bipartite matching is designed for each query set at different learning step, which includes advantages: 1) Avoiding the mutual interference between old and new knowledge. 2) Enabling adaptive background.

Standard bipartite matching between ground truth $\boldsymbol{y}^i$ and a predictions $\hat{\boldsymbol{y}}^{\sigma(i)}$ with index $\sigma(i)$ computes:

$$\hat{\sigma} = \underset{\sigma \in \Sigma_M}{\arg\min} \sum_{i}^{M} \mathcal{L}_{\text{match}}(\boldsymbol{y}^i, \hat{\boldsymbol{y}}^{\sigma(i)}), \tag{2}$$

where M is the number of predictions, the same as the number of queries. $\mathcal{L}_{\text{match}}(\boldsymbol{y}^i, \hat{\boldsymbol{y}}^{\sigma(i)})$ is a pair-wise matching cost [32]. For step t, we modify the matching to:

$$\hat{\sigma} = \underset{\sigma \in \Sigma_{M_t}}{\arg\min} \sum_{i}^{M_t} \mathcal{L}_{\text{match}}(\boldsymbol{y}_t^i, \hat{\boldsymbol{y}}_t^{\sigma(i)}), \tag{3}$$

where M_t, $\boldsymbol{y}_t^i$, and $\hat{\boldsymbol{y}}_t^{\sigma(i)}$ refer to the number of newly added query, the ground truth of new classes, and the corresponding prediction. Therefore, only the prediction of the newly added classes is supervised with annotation for the network. For the knowledge of old classes, we adopt *freezing* query embedding and query guided knowledge distillation (Sec. IV-B) to retain them.

B. Query Guided Knowledge Distillation

As shown in Fig. 2b, existing knowledge distillation methods [3], [4], [49] try to solve catastrophically forgetting via matching **global** statistics distribution at different feature levels between the old and current models. Given an embedding tensor $\boldsymbol{x}$, whose size is $H \times W \times C$. These methods usually focus on how to extract a better representation of the embedding tensor with a mapping function Φ (*e.g.,* , PCKD [49], POD [4]), then train the model $f_{\boldsymbol{\theta}}^t$ via minimizing the L2 distance between the two distribution:

$$\mathcal{L}(\Theta^t) = \frac{1}{L} \sum_{l=1}^{L} \left\| \Phi(\boldsymbol{x}_l^t) - \Phi(\boldsymbol{x}_l^{t-1}) \right\|^2, \tag{4}$$

where L denotes the number of embedding tensors. Actually, the **global** statistics match paradigm is crude and unreasonable with two drawbacks: 1) Directly matching will damage the representation learning of new classes, while $\boldsymbol{x}_l^{t-1}$ from the previous model $f_{\boldsymbol{\theta}}^{t-1}$ do not contain the representation. 2) Inter-class diversity and intra-class identity can not be captured for indiscriminate matching.

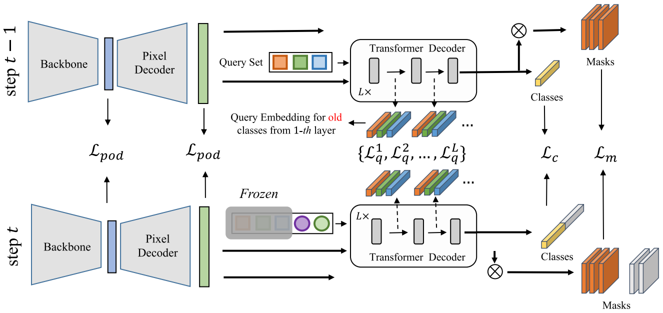


Fig. 3. **Illustration of Query Guided Knowledge Distillation**. Query G-KD mainly includes three parts: 1) Multi-Scale KD with local POD [3], [49] $\mathcal{L}_{pod}$ for feature from backbone and pixel-decoder; 2) Class/Instance-Aware Query KD $\{\mathcal{L}_q^1, \mathcal{L}_q^2, \dots, \mathcal{L}_q^L\}$ for query embedding vectors from multi-layers of transformer decoder; 3) Class/Instance-Aware Prediction KD for network prediction, *i.e.,* class $\mathcal{L}_c$ and mask $\mathcal{L}_m$.

To solve the two drawbacks, Class/Instance-Aware Query Guided Knowledge Distillation (Query G-KD) is proposed. Fig. 3 presents the whole illustration of Query G-KD, which includes three KD methods, *i.e.,* Multi-Scale KD with local POD, Class/Instance-Aware Query KD, and Class/Instance-Aware Prediction KD.

1) Class/Instance-Aware Query KD: At step t, given a C-dimensional learnable query embedding $\boldsymbol{q} \in \{Q_0, Q_1, \dots, Q_{t-1}\}$ that is responsible for a single *old* category $\mathcal{C}_{0:t-1}$, we can obtain L query embedding vectors $\{\boldsymbol{q}_1, \boldsymbol{q}_2 \dots \boldsymbol{q}_L\}$ from the output of each layer at transformer decoder for all L layers, as show in Fig. 3. In the training stage, The original query embedding $\boldsymbol{q}$ is frozen and extracted query embedding vectors $\{\boldsymbol{q}_1, \boldsymbol{q}_2 \dots \boldsymbol{q}_L\}$ are used for distillation. Then, the query knowledge distillation loss function $\mathcal{L}_q$ for the intermediate layers can be denoted as

$$\mathcal{L}_q(\Theta^t) = \frac{1}{M} \frac{1}{L} \sum_{j=0}^{M} \sum_{l=1}^{L} \left\| \boldsymbol{q}_{j,l}^t - \boldsymbol{q}_{j,l}^{t-1} \right\|^2, \tag{5}$$

where M is the number of all query embeddings $Q_{0:t-1}$ for *old* classes. And $\boldsymbol{q}_{j,l}^t$ and $\boldsymbol{q}_{j,l}^{t-1}$ refer to the student and teacher model from step t and $t-1$, respectively.

2) Class/Instance-Aware Prediction KD: To further retain the knowledge of old classes, prediction knowledge distillation is proposed for match classification and mask prediction between the student and teacher models. Given $t-1$ query groups $\{Q_0, Q_1, \dots, Q_{t-1}\}$ for old classes, we can obtain $t-1$ predicted class distribution vectors $\{\boldsymbol{c}_0, \boldsymbol{c}_1, \dots, \boldsymbol{c}_{t-1}\}$ from the output of class head before Softmax activation function. Therefore, we can match the distribution of old classes between two models with Kullback-Leibler (KL) divergence:

$$\mathcal{L}_c(\Theta^t) = \sum_{i=0}^{t-1} \boldsymbol{c}_i^{t-1} \log \frac{\boldsymbol{c}_i^{t-1}}{\boldsymbol{c}_i^t}, \tag{6}$$

where $\boldsymbol{c}_i^t$ and $\boldsymbol{c}_i^{t-1}$ refer to i-th class distribution prediction associated with query set $Q_i \in \{Q_0, Q_1, \dots, Q_{t-1}\}$ from step $t-1$ model (*i.e.,* teacher) and t model (*i.e.,* student), respectively. Similarly, the knowledge distillation of the predicted mask can be computed with the binary cross-entropy loss [28]

Algorithm 1 Training Pipeline of CisDQ

```
Input: $\mathcal{C}_0, \mathcal{I}_0$: the base class set and corresponding image set.
Input: $\{(\mathcal{C}_1, \mathcal{I}_1), (\mathcal{C}_2, \mathcal{I}_2), \cdots, (\mathcal{C}_T, \mathcal{I}_T)\}$ : the novel class set and
    corresponding image set for each incremental step.
Input: $\alpha$ : the learning rate.
1: Initialize model parameters $\boldsymbol{\theta}$ and the base query set $Q_0$ for
   CisDQ model $f$
2: Optimize $\{\boldsymbol{\theta}, Q_0\}$ with the Mask2former loss on image set $\mathcal{I}_0$
3: for incremental step $t \in \{1, \cdots, T\}$ do
4:     ▹ $Q = \{Q_i\}_{i=0}^{t-1} \cup Q_t$, Update the query set with random
5:       initialized novel query
6:     ▹ Freeze $\{Q_i\}_{i=0}^{t-1}$
7:     for each training step do
8:         ▹ Sample a mini-batch of training images $I$, $I \in \mathcal{I}_t$
9:         ▹ $\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}}(Q_{0:t}|I)$, get network predictions $\hat{\boldsymbol{y}}$
10:        ▹ Match the predictions $\hat{\boldsymbol{y}}$ to the ground truth $\boldsymbol{y} \in \mathcal{C}_t$
11:          with Equ. 3
12:        ▹ Calculate loss $\mathcal{L}$ with Equ. 8:
13:            $\mathcal{L} = \mathcal{L}_{\text{new}} + \lambda_1 \mathcal{L}_q + \lambda_2 \mathcal{L}_c + \lambda_3 \mathcal{L}_m + \lambda_4 \mathcal{L}_{pod}$
14:        ▹ Update $\{\boldsymbol{\theta}, Q_t\}$ with back propagation algorithm:
15:            $\{\boldsymbol{\theta}, Q_t\} = \{\boldsymbol{\theta}, Q_t\} - \alpha \nabla \mathcal{L}$
16:    end for
17: end for
Output: The trained CisDQ model $f$ with weights $\{\boldsymbol{\theta}, Q_{0:T}\}$
```

$\mathcal{L}_{ce}$ and dice loss [67] $\mathcal{L}_{dice}$:

$$\mathcal{L}_m(\Theta^t) = \lambda_c \frac{1}{M} \sum_{j=0}^{M} \mathcal{L}_{\text{ce}}(\boldsymbol{m}_j^t, \boldsymbol{m}_j^{t-1}) + \lambda_d \frac{1}{M} \sum_{j=0}^{M} \mathcal{L}_{\text{dice}}(\boldsymbol{m}_j^t, \boldsymbol{m}_j^{t-1}), \tag{7}$$

where M is the number of all query embeddings $Q_{0:t-1}$ for *old* classes. $\boldsymbol{m}_j^t$ and $\boldsymbol{m}_j^{t-1}$ denote the j-th predicted mask from step $t-1$ model (*i.e.,* teacher) and t model (*i.e.,* student), respectively. λ_c and λ_d are two weight parameters, similar to Mask2former [2].

C. Loss Function

The proposed pipeline mainly contains two losses, *i.e.,* manual annotation supervision loss, and knowledge distillation loss. The whole loss function can be formulated as Equation 8:

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{new}}}_{\text{New Classes}} + \underbrace{\lambda_1 \mathcal{L}_q + \lambda_2 \mathcal{L}_c + \lambda_3 \mathcal{L}_m + \lambda_4 \mathcal{L}_{pod}}_{\text{Distillation for Old Classes}}, \tag{8}$$

where $\mathcal{L}_{\text{new}}$ denotes the supervision from the available annotation of newly added classes, the same as that of Mask2former [2]. λ_1, λ_2, λ_3, λ_4 are the weight parameters, which are set to 1, 5, 300, 100, respectively. $\mathcal{L}_{pod}$ is the local POD loss [49], the same as PLOP [3], which is used to distill the features from backbone and pixel decoders.

D. Pseudo Code

We describe the training pipeline of CisDQ in Algorithm 1. We first train the model on the image set of the base classes (line 1-2). Then, for each incremental step, we add a set of new queries for the novel classes to the model and freeze the queries of the old classes (line 4-6). After that, we optimize the model with the proposed matching strategy (line 10) and loss function (line 12-15). Finally, we get the CisDQ model that is capable to segment objects of all classes.

V. Experiments

A. Experimental Setups

1) Semantic Segmentation: **Datasets.** Similar to previous works [3], [5], Pascal-VOC 2012 [74] (20 classes) and ADE20k [75] (150 classes) are used to evaluate our **CisDQ**. **Protocols.** Following works [3], [5], [76], the model is trained to segment new classes in multiple steps continually. And only new classes in the current step are labeled. And there are two different CSS settings: *Disjoint* and *Overlapped*. For step t, images of *Disjoint* only contain classes $\mathcal{C}^{1:t-1} \cup \mathcal{C}^t$ (old and new), while that of *Overlapped* can includes any classes $\mathcal{C}^{1:t-1} \cup \mathcal{C}^t \cup \mathcal{C}^{t+1:T}$ (old, new, and future). Therefore, the Overlapped setting is more challenging and realistic. In our experiments, we only focus on the result concerning Overlapped CSS. During the testing phase, the model need to segment all classes. Protocol setting following previous works [3], [4], *e.g.,* ADE 100-50, 100-10, 50-50, and 100-5, which consists in learning 100 classes followed by 50 class (2 steps), 100 classes followed by five times 10 classes (6 steps), 50 classes followed by two times 50 classes (3 steps), and 100 classes followed by ten times 10 classes (11 steps).

2) Instance Segmentation: **Datasets.** Following general instance segmentation [2], [30], COCO [77] (80 classes) and ADE20k [75] (100 classes) are used to evaluate our **CisDQ**. **Protocols.** Different from CCS, Continual Instance Segmentation (CIS), as one new task, need the first definition for experiment settings. As for COCO [77], we give three-step settings, *i.e.,* 40-40 (2 steps), 40-8 (6 steps), and 40-4 (11 steps). For ADE20k, similar to the class split of CCS, we provide three step settings concerning two steps (50-50), six steps (50-10), and eleven steps (50-5). The detailed classes split for each step and related experiments for ADE20k are provided in Table III. Similar to CCS, we focus on *Overlapping* setting, which is more realistic and challenging.

3) Implementation Details: The most experiment setting of the experiments all follow Mask2former [2], *i.e.,* backbone, transformer, pixel decoder, AdamW optimizer with an initial learning rate of 0.0001. And 8 Tesla V100 GPUs are used for the experiment. For *semantic* segmentation, the number of the query is the same as the that of class during continual learning, *e.g.,* ADE 100-50 (2 steps) require 100 queries to learn 1-100 classes at step 0, then increasing 50 queries to learn 101-150 classes at step 1. For *instance* segmentation, the number of the query is set to 2.5 times of that of class during continual learning, COCO requires 200 queries no matter which settings. COCO 40-20 (3 steps) requires 100, 50, 50 queries for 0-th, 1-th, 2-th steps, respectively.

4) Compared With iFSIS: Tab. I compares continual instance segmentation (CIS) with other related problem, *i.e.,* incremental few-shot instance segmentation [65], [66]. Different from continual segmentation, iFSIS only incremental learning on base and novel classes with just two steps, and the training data for new classes is limited and few.

TABLE I

COMPARISON OF RELATED TASK. 'iFSIS' AND 'CIS' REFER TO INCREMENTAL FEW-SHOT INSTANCE SEGMENTATION AND CONTINUAL INSTANCE SEGMENTATION, RESPECTIVELY

Settings	Step Number	Few Shot for New Class
iFSIS	Two-Step (Step 0, 1)	✓
CIS	Multi-Step (Step 0, 1, ..., n)	

TABLE II

CLASS ORDER, FOR INSTANCE, SEGMENTATION WITH 80 CLASSES ON COCO 2017

Seq.	80 Instance Classes on COCO 2017
1-8	person, bicycle, car, motorcycle, airplane, bus, train, truck
9-16	boat, traffic light, fire hydrant, stop sign, parking meter, bench, bird, cat
17-24	dog, horse, sheep, cow, elephant, bear, zebra, giraffe
25-32	backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard
33-40	sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket, bottle
41-48	wine glass, cup, fork, knife, spoon, bowl, banana, apple
49-56	sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake
57-64	chair, couch, potted plant, bed, dining table,toilet, tv, laptop
65-72	mouse, remote, keyboard, cell phone, microwave, oven, toaster, sink
72-80	refrigerator, book, clock, vase, scissors, teddy bear, hair drier, toothbrush

TABLE III

CLASS ORDER, FOR INSTANCE, SEGMENTATION WITH 100 CLASSES ON ADE 20K

Seq.	100 Instance Classes on ADE 20k
1-10	bed, windowpane, cabinet, person, door, table, curtain, chair, car, painting,
11-20	sofa, shelf, mirror, armchair, seat, fence, desk, wardrobe, lamp, bathtub,
21-30	railing, cushion, box, column, signboard, chest of drawers, counter, sink, fireplace, refrigerator,
31-40	stairs, case, pool table, pillow, screen door, bookcase, coffee table, toilet, flower, book ,
41-50	bench, countertop, stove, palm, kitchen island, computer, swivel chair, boat, arcade machine, bus,
51-60	towel, light, truck, chandelier, awning, streetlight, booth, television receiver, airplane, apparel,
61-70	pole, bannister, ottoman, bottle, van, ship, fountain, washer, plaything, stool,
71-80	barrel, basket, bag, minibike, oven, ball, food, step, trade name, microwave,
81-90	pot, animal, bicycle, dishwasher, screen, sculpture, hood, sconce, vase, traffic light,
91-100	tray, ashcan, fan, plate, monitor, bulletin board, radiator, glass, clock, flag

5) Protocols for Continual Instance Segmentation Datasets: COCO 2017. Different from continual semantic segmentation, Continual Instance Segmentation (CIS), as one new task, needs the first definition for experiment settings. As for COCO 2017 [77], we give three-step settings, *i.e.,* 40-40 (2 steps), 40-8 (6 steps), and 40-4 (11 steps). And Table. II provides the corresponding class order. For ADE20k, similar to the class split of CCS, we provide three-step settings concerning two steps (50-50), six steps (50-10), and eleven steps (50-5). The detailed class order for splitting is provided in Table. III. Similar to CCS, all experiments concerning instance segmentation task focus on *Overlapping* setting, which is more realistic and challenging.

B. Continual Semantic Segmentation

1) PASCAL VOC 2012: Table. V presents the experimental results of the last step for three continual learning settings. Compared with previous methods based on deeplab v3, our **CISDQ** achieves obvious mIoU improvements, especially for long-term steps setting, *e.g.,* VOC 10-1 (11 steps) and 15-1 (6 steps). For a fair comparison, the performance of SSUL [5] is not the main reference, while it uses the extra Salient Object Segmentation model pre-trained on MSRA-B dataset [72]). Actually, the Salient object segmentation model usually can segment and obtain competitive pseudo label effectively, especially for VOC 2012, where the image domain is simple and only support 20 classes. And we also re-implementation PLOP [3] with Mask2former framework. We do not re-implementation RCIL [4] and SSUL [5] on Mask2former architecture due to the mismatched re-parameterization on transformer architecture from RCIL and using extract model of SSUL. Overall, our **CISDQ** achieves state-of-the-art performance with obvious improvements no matter for the old or new classes. Besides, we also provide more high-quality visualization results for semantic segmentation tasks on ADE 20k, as shown in Fig. 4.

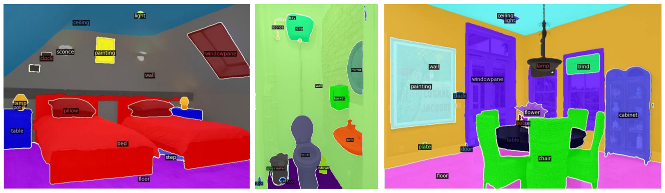


Fig. 4. **More visualization of CISDQ for semantic segmentation on ADE 20k `val`. CISDQ** presents high-quality results.

2) ADE20k: Table. IV presents the experimental results of the last step for four continual learning settings on ADE 20K. On different continual learning tasks, *i.e.,* 100-50, 100-10 and 50-50, 100-5, our method all achieves state-of-the-art results, especially for the long step tasks, *i.e.,* 100-10 (6 steps) and 100-5 (11 steps), achieving 4.4% and 2.9% mIoU improvement over that of previous SOTA method, respectively. Our algorithm demonstrates significant improvements compared to the ten methods based on Deeplab v3 or Mask2Former, substantiating the effectiveness of our approach. For fair comparison, the performance of SSUL [5] and MicroSeg-M [63] is not the main reference, while it uses the extra Salient Object Segmentation model pre-trained on MSRA-B dataset [72]). And we also re-implementation MiB [62] and PLOP [3] with Mask2former framework. We do not re-implementation RCIL [4] and SSUL [5] on Mask2former due to the mismatched re-parameterization on transformer architecture of RCIL and the extra model of SSUL.

C. Continual Instance Segmentation

1) COCO2017: To verify the effectiveness of **CISDQ** for instance segmentation, we provide corresponding experiments on COCO [77] for three different continual tasks, *i.e.,* COCO 40-40, 40-8, and 40-4, as shown in Table. VI. **CISDQ** presents competitive performance, around 1.0% mAP improvement over previous methods. It is worth noting that **CISDQ** presents a better performance for newly added classes, power from more reasonable KD for class/instance level.

2) ADE 20k: We also provide the results for continual instance segmentation with three continual settings on ADE 20k, as shown in Fig. VII. We implement PLOP [3] on Mask2former architecture for continual instance segmentation, while there is no one continual instance segmentation method for comparison. **CISDQ** achieves obvious improvements over PLOP, with 6.1%, 6.7%, and 2.8% mAP for the three different settings, respectively. Besides, we also provide some visualizations for continual instance segmentation on COCO 2017, as shown in Fig. 6.

D. Ablation Study

Incremental query queue and independent bipartite match, as two indispensable points, enable our dynamic

TABLE IV

The Final mIoU(%) of Semantic Segmentation on the ADE20K dataset. ⋆ Refers to Our Re-Implementation Performance. In Gray Denotes Joint Training With All Classes or Using Extra Model (*i.e.*, SSUL Using DSS [71] Pretrained on MSRA-B Dataset [72])

Method	Base Network	Backbone	ADE 100-50 (2 steps)			ADE 100-10 (6 steps)			ADE 50-50 (3 steps)			ADE 100-5 (11 steps)		
			1-100	*101-150*	*all*	*1-100*	*101-150*	*all*	*1-50*	*51-150*	*all*	*1-100*	*101-150*	*all*
Joint Training	Deeplab V3	ResNet-101	44.3	28.2	38.9	44.3	28.2	38.9	51.1	33.3	38.9	44.3	28.2	38.9
ILT [68]	Deeplab V3	ResNet-101	18.3	14.4	17.0	0.1	3.1	1.1	3.5	12.9	8.7	0.1	1.3	0.5
MiB [62]	Deeplab V3	ResNet-101	40.5	17.2	32.8	38.2	11.1	29.2	45.6	21.0	29.3	36.0	5.7	26.0
PLOP [3]	Deeplab V3	ResNet-101	41.9	14.9	32.9	40.5	13.6	31.6	48.8	21.0	30.4	39.1	7.8	28.8
RCIL [4]	Deeplab V3	ResNet-101	42.3	18.8	34.5	39.3	17.6	32.1	48.3	25.0	32.5	38.5	11.5	29.6
MiB+EWF [69]	Deeplab V3	ResNet-101	41.2	21.3	34.6	41.5	-	33.2	-	-	-	41.4	13.4	32.1
SSUL [5]	Deeplab V3	ResNet-101	41.3	18.0	33.6	40.2	18.8	33.1	48.4	20.2	29.6	39.9	17.4	32.5
Joint Training ⋆	Mask2former	ResNet-50	49.8	36.9	45.5	49.8	36.9	45.5	55.9	40.3	45.5	49.8	36.9	45.5
CoMFormer [70]	Mask2former	ResNet-101	44.7	26.2	38.4	40.6	15.6	32.3	-	-	-	39.5	13.6	30.9
MiB ⋆	Mask2former	ResNet-50	47.6	27.8	41.0	40.6	16.3	32.5	52.7	30.8	38.1	38.0	12.2	29.4
PLOP ⋆	Mask2former	ResNet-50	48.2	28.0	41.5	44.3	18.6	35.7	55.5	32.0	39.9	39.2	13.2	30.5
MicroSeg [63]	Mask2former	ResNet-101	40.2	18.8	33.1	41.5	21.6	34.9	48.6	24.8	32.9	40.4	20.5	33.8
MicroSeg-M [63]	Mask2former	ResNet-101	43.4	20.9	35.9	43.7	22.2	36.6	49.8	22.0	31.4	43.6	22.4	36.6
CiSDQ (ours)	Mask2former	ResNet-50	**48.9**	**28.2**	**42.0**	**47.8**	**24.6**	**40.1**	**55.7**	**33.6**	**41.0**	**46.2**	**17.6**	**36.7**

TABLE V

The mIoU(%) of Semantic Segmentation on the Pascal VOC 2012 dataset. ⋆ Refer to Our Re-Implementation Performance. In Gray Denotes Joint Training With All Classes or Using Extra Model (*i.e.*, SSUL Using DSS [71] Pretrained on MSRA-B Dataset [72])

Method	Base Network	Backbone	VOC 15-1 (6 steps)			VOC 15-5 (2 steps)			VOC 10-1 (11 steps)		
			0-15	*16-20*	*all*	*0-15*	*16-20*	*all*	*0-10*	*11-20*	*all*
Joint Training (offline)	Deeplab V3	ResNet-101	79.8	72.3	77.4	79.8	72.4	77.4	78.4	76.4	77.4
LwF-MC [73]	Deeplab V3	ResNet-101	6.4	8.4	6.9	58.1	35.0	52.3	4.7	5.9	5.0
ILT [68]	Deeplab V3	ResNet-101	8.8	8.0	8.6	67.1	39.2	60.5	7.2	3.7	5.5
MiB [62]	Deeplab V3	ResNet-101	35.1	13.5	29.7	75.5	49.4	69.0	12.2	13.1	12.6
PLOP [3]	Deeplab V3	ResNet-101	65.1	21.1	54.6	75.7	51.7	70.1	44.0	15.5	30.5
RCIL [4]	Deeplab V3	ResNet-101	70.6	**23.7**	59.4	78.8	**52.0**	72.4	55.4	15.1	34.3
SSUL [5]	Deeplab V3	ResNet-101	77.3	36.6	67.6	77.8	50.1	71.2	71.3	46.0	59.3
Joint Training (offline) ⋆	Mask2former	ResNet-50	80.0	75.2	78.8	80.0	75.2	78.8	78.9	78.8	78.8
PLOP ⋆	Mask2former	ResNet-50	71.3	12.5	57.3	78.2	28.4	66.3	56.9	14.5	36.7
CiSDQ (ours)	Mask2former	ResNet-50	**77.9**	13.2	**62.5**	**80.5**	49.3	**73.1**	**73.2**	**15.5**	**45.7**
CiSDQ (ours)	Mask2former	ResNet-101	78.2	13.6	62.8	80.6	49.5	73.2	73.9	15.8	46.2
CiSDQ (ours)	Mask2former	Swin-B	79.7	14.9	64.3	84.0	55.1	77.1	74.6	15.9	46.7

TABLE VI

The Final AP(%) of Instance Segmentation on COCO *val2017* With 80 categories. Mask2former With ResNet50 Is Used as the Base Network. 'offline' Refers to Training With All Data and Classes. ⋆ Refers to Our Re-Implementation Performance

Method	COCO 40-40 (2 steps)			COCO 40-8 (6 steps)							COCO 40-4 (11 steps)		
	1-40	*41-80*	*mAP*	*1-40*	*41-48*	*49-56*	*57-64*	*65-72*	*73-80*	*mAP*	*1-40*	*41-80*	*mAP*
Joint Training (offline)	35.6	51.8	43.7	35.6	47.0	59.2	53.8	52.6	46.2	43.7	35.6	51.8	43.7
PLOP ⋆	28.0	40.1	34.1	**24.4**	16.4	28.0	22.3	25.1	16.3	23.0	16.5	11.3	13.9
CiSDQ	**28.8**	**41.8**	**35.3**	24.2	**17.9**	**29.4**	**24.8**	**27.6**	**19.8**	**24.1**	**18.2**	**11.7**	**15.0**

TABLE VII

The Final AP(%) of Instance Segmentation on the ADE20K dataset. ⋆ Refers to Our Re-Implementation Performance. ResNet50 as the Backbone Is Used

Method	ADE 50-50 (2 steps)			ADE 50-10 (6 steps)							ADE 50-5 (11 steps)		
	1-50	*51-100*	*all*	*1-50*	*51-60*	*61-70*	*71-80*	*81-90*	*91-100*	*all*	*1-50*	*51-100*	*all*
Joint Training (offline)	32.2	20.6	26.4	32.2	24.6	14.2	16.9	26.7	20.5	26.4	32.2	20.6	26.4
PLOP ⋆	23.4	11.8	17.6	13.1	6.4	2.1	7.2	11.1	4.9	9.7	12.9	2.5	7.7
CiSDQ (ours)	31.2	13.8	23.7	26.2	7.1	3.9	7.9	9.8	5.0	16.4	18.1	2.9	10.5

query and adaptive background. Table. VIII presents the effect of the two components. Without the independent match, the incremental query just increases the number of queries, which can not decouple the learning of old and new classes.

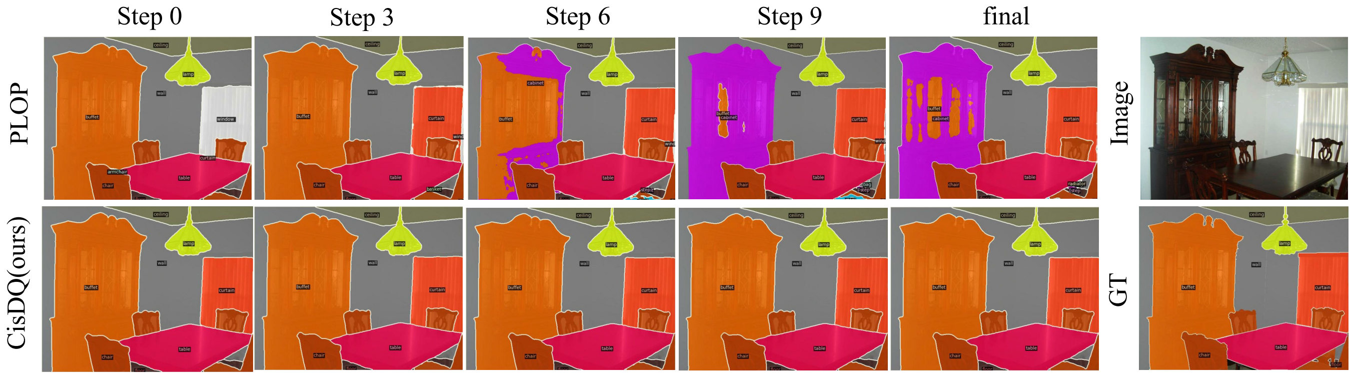


Fig. 5. **Visualization of PLOP and CISDQ for *100-5* (11 steps) on ADE 20k `val`. CISDQ** presents more robust performance for remaining the knowledge of old classes (*e.g.,* `buffet`), while PLOP [3] suffer from the catastrophic forgetting.

TABLE VIII

CISDQ ABLATIONS. WE PERFORM ABLATIONS ON ADE20K `VAL` FOR CONTINUAL SEMANTIC SEGMENTATION TASK

Incremental Query	Independent Match	1-100	101-150	*all*
		39.2	13.2	30.5
✓		39.6 (+0.4)	15.3 (+2.1)	31.5(+1.0)
✓	✓	46.2 (+6.6)	17.6 (+2.3)	36.7 (+5.2)

(a) **Dynamic Query Queue**. Experiments are conducted on *100-5* (11 steps) setting for semantic segmentation (mAP) task on ADE20k `val`.

Method	Base Network	Backbone	Params	Flops	FPS	*mIoU*
SSUL	Deeplab V3	ResNet-101	62.7	255.1	14.1	33.1
CiSDQ	Mask2former	ResNet50	44.0	71.0	9.7	40.1

(b) **Base Network.** Experiments are conducted on *100-10* setting for semantic segmentation (mIoU) task on ADE20K `val`.

Prediction-KD	Pod-KD	Query-KD	1-100	101-150	*all*
			0.7	2.7	1.2
✓			35.5	15.1	28.7
✓		✓	41.3	17.9	33.5
✓	✓		38.7	15.6	31.0
✓	✓	✓	46.2	17.6	36.7

(c) **Query-KD and Prediction-KD.** Experiments are conducted for *100-5* setting on ADE20k `val`.

Query Frozen	1-100	101-150	*all*
	44.2	17.3	35.2
✓	46.2	17.6	36.7

(d) **Frozen for Old Query Embedding.** Experiments are conducted on *100-5* (6 steps) overlapped setting for semantic segmentation (mIoU) task on ADE20k `val`.

100-5	Queries	Params	FPS	*mIoU*
Step 0 (*1-100*)	100	43.915	11.0	32.2
Step 1 (*1-110*)	110	43.916	10.8	32.3
Step 5 (*1-150*)	150	43.920	9.7	36.7

(e) **Increasing Params for Dynamic Query.** Experiments are conducted on *100-5* setting for semantic segmentation (mIoU) task on ADE20K `val`.

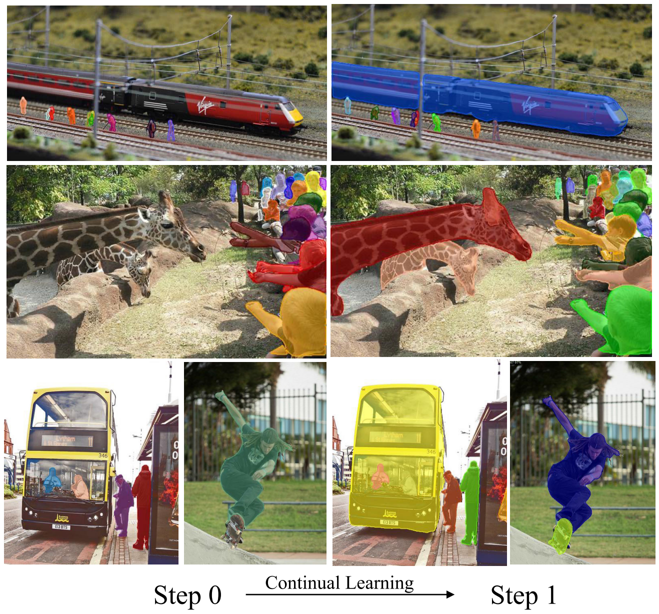


Fig. 6. **Visualization of CISDQ for continual instance segmentation on COCO 2017 `val`. CISDQ** presents robust performance on learning new classes and remaining old knowledge.

Therefore, with the only incremental query, **CISDQ** achieves a little improvement, around 1.0% mIoU. By contrast, the combination of two components brings obvious improvement, up to 5.2% mIoU over the baseline.

Query-KD and Prediction-KD, two key components for our Query Guided Knowledge Distillation. Table. VIIIc presents the effect for the two components. Baseline, without the three KDs, training with new classes directly, presents an unacceptable result 1.2% mIoU. Prediction-KD, Pod-KD, and Query-KD bring different gains for the final results, *i.e.,* 27.5%, 2.3%, 5.7%, respectively. Although Pod-KD damages the representation learning of new classes due to the unbalanced matching in Fig. 2b, the positive impact for remaining the knowledge of old classes is not neglectable. Unlike our Query-Guided KD, Pod-KD forces on the features from the backbone and pixel decoder, thus we adopt it in Equ. 8. Actually, any KDs will damage the performance of newly added classes. Compared with Pod-KD, Query-KD shows better performance on newly added classes (*101-150* classes), where 17.9% *v.s* 15.6% mIou on 3-*th* and 4-*th* lines. Meanwhile, Query-KD also gives a competitive performance on old classes, as shown in Fig. 5.

1) Frozen for Query of Old Class: Except for knowledge distillation, query embedding frozen also help to remain the performance of old classes, as shown in Table. VIIId. With query frozen, the performance of old classes (*1-100* classes) shows an obvious improvement with 1.5% mIoU.

TABLE IX

The Ablation Study for Model Generalization During Continual learning. We Conducted Cross-Dataset Validation to Investigate the Generalization Capability of the Model. In the ADE 50-50 (3 Steps) Setting, the 'Bus' and 'Boat' Categories Are Considered as New Classes, Requiring the Model to Learn Them in the 1-Th Step

Class	**Train Set**	**Test Set**	**ADE 50-50 (3 steps,mIoU/%)**			**ADE 100-10 (6 steps,mIoU/%)**					
			0-th step	1-th step	2-th step	0-th step	1-th step	2-th step	3-th step	4-th step	5-th step
Person	ADE20K	VOC 2012	76.7	76.4	75.6	77.2	76.3	76.7	76.8	75.5	75.0
Car	ADE20K	VOC 2012	74.1	75.5	75.1	75.6	75.8	75.7	78.6	79.4	79.5
Bus	ADE20K	VOC 2012	0	68.7	68.2	62.0	61.3	60.1	52.3	50.0	50.8
Boat	ADE20K	VOC 2012	0	33.4	37.7	26.0	26.5	27.2	28.7	27.7	27.4

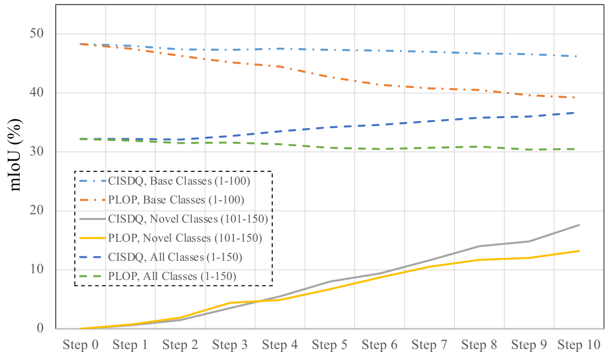


Fig. 7. **Compared with PLOP [3] for semantic segmentation task on ADE 20k 100-5 (11 steps). CISDQ** presents better performance.

2) Computation Cost From Increasing Query Embedding: We also study the extra computation cost from increasing queries in Table. VIIIe. For *100-5* (11 steps) setting, the number of queries will increase from 100 to 150 to represent the newly added classes. And the increased 50 queries only bring around 0.005 M extra parameters, while the performance shows an obvious improvement. Compared with Deeplab V3 based methods, mask2former-based framework (ours) shows a big advantage with low computation cost (fewer parameters, smaller flops, and faster speed) and better performance.

3) Base Framework: Table. VIIIb gives a comparison for the two different base networks. Mask2former is a better choice for us to apply continual learning to image segmentation, while the parameters, flops, and FPS are all acceptable.

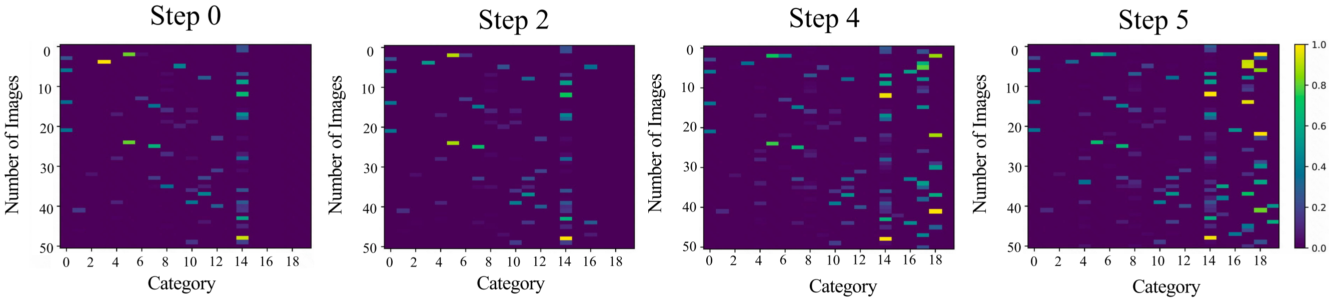
4) Comparison Between PLOP: Fig. 7 presents the comparison between PLOP [3] and **CISDQ** for semantic segmentation task on ADE 20k 100-5 setting. As for the learning of newly added classes, **CISDQ** shows better performance over PLOP, which is mainly caused by two aspects: 1) Different queries decouple the learning of new classes and remaining old knowledge; 2) Query-Guided KD performs a more precise distillation matching without damaging the data distribution of new classes. Similarly, **CISDQ** presents less performance loss for old classes during the continual learning of newly added classes process.

5) Ablation Study for Model Generalization: Table IX present the ablation study for model generalization during continual learning. We conducted cross-dataset evaluation to study the model's generalization. Due to variations in the number and distribution of categories across different datasets, such as the inclusion of the 'refrigerator' category in ADE 20K, which is not present in the VOC 2012 dataset, we opted to explore specific categories. In fact, the generalization variations of the model differ for different categories during the continual learning process. For example, the generalization of the 'person' category tends to decrease, while the 'car' category shows improvement. We attribute these changes to a balance between the forgetting of knowledge related to old classes and the decrease in false positives. In the continual learning process, the forgetting of knowledge related to old classes is a recognized challenge that can lead to a decline in the performance of these classes. However, as the number of categories increases during continual learning, it can also assist an old class in excluding certain regions that are prone to false positives. As a result, different categories and experimental settings may yield varied results. Overall, the generalization remains stable. It is essential to emphasize that continual learning is not aimed at enhancing the model's generalization but rather enabling the model to dynamically update its knowledge and extend its ability to segment new classes.

6) Visualization of the Output Tensor: In order to gain a deeper understanding of the continual learning process and the changes in the model, we randomly selected 50 images from the VOC 2012 test set. Then we extracted and visualized the multiplication of the pixel decoder and transformer decoder outputs. Given 50 images, we obtain corresponding tensor outputs of size $50 \times 20 \times w \times h$, where 20 refers to the number of category (Excluding the background category). For ease of visualization and examining activations for each category, we first use average pooling to transform the tensor from $50 \times 20 \times w \times h$ to $50 \times 20 \times 1 \times 1$. Subsequently, we normalize the tensor and visualize the results as shown in Figure 8. At 0-th step, it is evident that the tensor lacks activations for the last five classes, such as 'pottedplant' and 'sheep.' However, as continual learning progresses and the model learns new classes, the tensor exhibits activations for the new classes, allowing for predictions of these novel categories.

VI. Limitations and Future Works

Although CisDQ can achieve good performance on continual semantic segmentation (CSS), its performance on continual instance segmentation (CIS) is still not satisfactory. Compared to CSS, CIS requires more queries as instances from the same class can only be distinguished by different queries. In our future work, we will explore a way to assign queries to the old and novel classes more efficiently, so that classes from



Categories for *0*-th Step : aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person
Category for *1*-th Step : pottedplant Category for *2*-th Step : sheep Category for *3*-th Step : sofa Category for *4*-th Step : train Category for *5*-th Step : tvmonitor

Fig. 8. **Visualization of the output tensor of the decoder for 50 randomly sampled images on the VOC 2012 dataset.** Throughout the continual learning process, the model exhibits activations for novel classes, and there is no forgetting of the feature for old classes.

each incremental step can get enough queries to achieve high performance on CIS.

VII. Conclusion

In this paper, we propose a simple, yet effective Continual Image Segmentation method with incremental Dynamic Query, named **CISDQ**, which decouples the representation learning of both old and new knowledge with lightweight query embedding. **CISDQ** also includes a novel class/instance-aware Query Guided Knowledge Distillation strategy to overcome catastrophic forgetting by capturing the inter-class diversity and intra-class identity. Meanwhile, we introduce the continual learning to instance segmentation task, which is more challenging. Experimental results show that our **CISDQ** achieves SOTA performance, with up to 10% mIoU improvement over previous methods.

References

[1] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[2] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1280–1289.

[3] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "PLOP: Learning without forgetting for continual semantic segmentation," 2020, *arXiv:2011.11390*.

[4] C.-B. Zhang, J.-W. Xiao, X. Liu, Y.-C. Chen, and M.-M. Cheng, "Representation compensation networks for continual semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7043–7054.

[5] S. Cha et al., "SSUL: Semantic segmentation with unknown label for exemplar-based class-incremental learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 10919–10930.

[6] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Sci.*, vol. 7, no. 2, pp. 123–146, Jun. 1995.

[7] R. French, "Catastrophic forgetting in connectionist networks," *Trends Cognit. Sci.*, vol. 3, no. 4, pp. 128–135, Apr. 1999.

[8] W. Wu et al., "End-to-end video text spotting with transformer," 2022, *arXiv:2203.10539*.

[9] W. Wu et al., "A bilingual, openworld video text dataset and end-to-end video text spotter with transformer," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round)*, 2021. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021

[10] Y. Zhao, Y. Cai, W. Wu, and W. Wang, "Explore faster localization learning for scene text detection," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 156–161.

[11] C. Gong et al., "Curiosity-driven and victim-aware adversarial policies," in *Proc. 38th Annu. Comput. Secur. Appl. Conf.*, Dec. 2022, pp. 186–200.

[12] M. S. Allili, D. Ziou, N. Bouguila, and S. Boutemedjet, "Image and video segmentation by combining unsupervised generalized Gaussian mixture modeling and feature selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1373–1377, Oct. 2010.

[13] L. Salgado, N. Garcia, J. M. Menendez, and E. Rendon, "Efficient image segmentation for region-based motion estimation and compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 7, pp. 1029–1039, 2000.

[14] T. Ida and Y. Sambonsugi, "Image segmentation using fractal coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 6, pp. 567–570, 1995.

[15] H. Lu, J. C. Woods, and M. Ghanbari, "Binary partition tree for semantic object extraction and image segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 3, pp. 378–383, Mar. 2007.

[16] S. Sun, D. R. Haynor, and Y. Kim, "Semiautomatic video object segmentation using vsnakes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 75–82, Jan. 2003.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[18] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr, "Higher order conditional random fields in deep neural networks," in *Proc. 14th Eur. Conf.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 524–540, doi: 10.1007/978-3-319-46448-0.

[19] S. Zheng et al., "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.

[20] D. Zhang, H. Zhang, and J. Tang, "Causal intervention for weakly-supervised semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 655–666.

[21] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, "Self-regulation for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6933–6943.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[24] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," 2015, *arXiv:1506.04579*.

[25] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[26] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[27] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12077–12090.

[28] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17864–17875.

[29] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proc. 13th Eur. Conf.* Zurich, Switzerland: Springer, Sep. 2014, pp. 297–312, doi: 10.1007/978-3-319-10593-2.

[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1440–1448.

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. 16th Eur. Conf.* Glasgow, U.K.: Springer, 2020, pp. 213–229, doi: 10.1007/978-3-030-58580-8.

[33] Y. Zhao, Q. Ye, W. Wu, C. Shen, and F. Wan, "Generative prompt model for weakly supervised object localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6351–6361.

[34] W. Wu, Y. Zhao, M. Z. Shou, H. Zhou, and C. Shen, "DiffuMask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 1206–1217.

[35] W. Wu et al., "DatasetDM: Synthesizing data with perception annotations using diffusion models," 2023, *arXiv:2308.06160*.

[36] D. Yu et al., "Contrastive correlation preserving replay for online continual learning," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jun. 12, 2023, doi: 10.1109/TCSVT.2023.3285221.

[37] X. Fu, J. Xiao, Y. Zhu, A. Liu, F. Wu, and Z.-J. Zha, "Continual image deraining with hypergraph convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9534–9551, Aug. 2023.

[38] Z. Le et al., "UIFGAN: An unsupervised continual-learning generative adversarial network for unified image fusion," *Inf. Fusion*, vol. 88, pp. 305–318, Dec. 2022.

[39] M. Zhou et al., "Image de-raining via continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4905–4914.

[40] H. Xu, J. Ma, J. Jiang, X. Guo, and H. Ling, "U2Fusion: A unified unsupervised image fusion network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 502–518, Jan. 2022.

[41] R. Yan, L. Xie, X. Shu, and J. Tang, "Interactive fusion of multi-level features for compositional activity recognition," 2020, *arXiv:2012.05689*.

[42] R. Yan, L. Xie, J. Tang, X. Shu, and Q. Tian, "Social adaptive module for weakly-supervised group activity recognition," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Springer, Aug. 2020, pp. 208–224.

[43] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivat.*, vol. 24, pp. 109–165, Dec. 1989.

[44] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8214–8223.

[45] E. Belouadah and A. Popescu, "IL2M: Class incremental learning with dual memory," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 583–592.

[46] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 8, pp. 6993–7001.

[47] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 532–547.

[48] A. Cheraghian, S. Rahman, P. Fang, S. K. Roy, L. Petersson, and M. Harandi, "Semantic-aware knowledge distillation for few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2534–2543.

[49] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "PODNet: Pooled outputs distillation for small-tasks incremental learning," in *Proc. 16th Eur. Conf.* Glasgow, U.K.: Springer, Aug. 2020, pp. 86–102, doi: 10.1007/978-3-030-58580-8.

[50] Y. Xiang, Y. Fu, P. Ji, and H. Huang, "Incremental learning using conditional adversarial networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6618–6627.

[51] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, "Adversarial continual learning," in *Proc. 16th Eur. Conf.* Glasgow, U.K.: Springer, 2020, pp. 386–402, doi: 10.1007/978-3-030-58580-8.

[52] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[53] M. Xue, H. Zhang, J. Song, and M. Song, "Meta-attention for ViT-backed continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 150–159.

[54] A. Ashok, K. J. Joseph, and V. Balasubramanian, "Class-incremental learning with cross-space clustering and controlled transfer," 2022, *arXiv:2208.03767*.

[55] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "DyTox: Transformers for continual learning with dynamic token eXpansion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9275–9285.

[56] M. Boschini et al., "Transfer without forgetting," 2022, *arXiv:2206.00388*.

[57] Z. Huang et al., "Half-real half-fake distillation for class-incremental semantic segmentation," 2021, *arXiv:2104.00875*.

[58] S. Yan, J. Zhou, J. Xie, S. Zhang, and X. He, "An EM framework for online incremental learning of semantic segmentation," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3052–3060.

[59] F. Cermelli, M. Mancini, S. R. Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9230–9239.

[60] H. Zhao, F. Yang, X. Fu, and X. Li, "RBC: Rectifying the biased context in continual semantic segmentation," in *Proc. 17th Eur. Conf.* Tel Aviv, Israel: Springer, Oct. 2022, pp. 55–72, doi: 10.1007/978-3-031-20083-0.

[61] E. Zheng, Q. Yu, R. Li, P. Shi, and A. Haake, "A continual learning framework for uncertainty-aware interactive image segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 7, pp. 6030–6038.

[62] F. Cermelli, M. Mancini, S. R. Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9230–9239.

[63] Z. Zhang, G. Gao, Z. Fang, J. Jiao, and Y. Wei, "Mining unseen classes via regional objectness: A simple baseline for incremental segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 24340–24353.

[64] Y. Qiu et al., "SATS: Self-attention transfer for continual semantic segmentation," 2022, *arXiv:2203.07667*.

[65] D. A. Ganea, B. Boom, and R. Poppe, "Incremental few-shot instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1185–1194.

[66] K. Nguyen and S. Todorovic, "IFS-RCNN: An incremental few-shot instance segmenter," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7000–7009.

[67] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

[68] U. Michieli and P. Zanuttigh, "Incremental learning techniques for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3205–3212.

[69] J. Xiao, C. Zhang, J. Feng, X. Liu, J. van de Weijer, and M. Cheng, "Endpoints weight fusion for class incremental semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7204–7213.

[70] F. Cermelli, M. Cord, and A. Douillard, "CoMFormer: Continual learning in semantic and panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 3010–3020.

[71] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5300–5309.

[72] T. Liu et al., "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.

[73] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[74] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[75] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5122–5130.

[76] F. Cermelli, M. Mancini, S. R. Bulò, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9230–9239.

[77] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf.* Zurich, Switzerland: Springer, Sep. 2014, pp. 740–755, doi: 10.1007/978-3-319-10593-2.

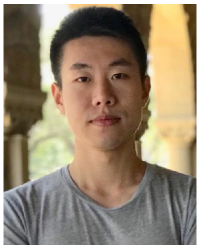
Weijia Wu received the B.E. degree in electronic information engineering from Wenzhou University, Wenzhou, China, in 2018. He is currently pursuing the Ph.D. degree with Zhejiang University. He is a Visiting Student with the National University of Singapore. His research interests include quite relevant to long-form video understanding, scene text detection, recognition, and cross-modal video-and-language retrieval.

Yuzhong Zhao (Member, IEEE) received the B.S. degree from Peking University, Beijing, China, in 2020. He is currently pursuing the M.E. degree with the University of Chinese Academy of Sciences, Beijing. His research interests include computer vision, scene text detection, and recognition.

Zhuang Li received the B.E. and M.S. degrees in computer application from Northeastern University, Shenyang, China. He is currently an Algorithm Engineer with the MMU Vision Center, Kuaishou Technology. His research interests include image and video generation, text detection, and recognition.

Lianlei Shan received the B.E. degree from the Shandong University of Science and Technology in 2018. He is currently pursuing the Ph.D. degree with the School of Computer and Control Engineering, University of Chinese Academy of Sciences. His research interests include remote sensing image processing and computer vision.

Hong Zhou received the B.E. degree in measurement technology and instrument and the Ph.D. degree in instrument science and technology from Zhejiang University, Hangzhou, China, in 1995 and 2000, respectively. He is currently a Professor with the College of Biomedical Engineering & Instrument Science, Zhejiang University. His research interests include video analysis technology and embedded systems.



Mike Zheng Shou (Member, IEEE) received the Ph.D. degree from the Columbia University in the City of New York, under the supervision of Prof. Shih-Fu Chang. He was a Research Scientist with Facebook AI in Bay Area. He is currently a Tenure-Track Assistant Professor with the National University of Singapore. He is a fellow of the National Research Foundation (NRF) Singapore. He was awarded the Wei Family Private Foundation Fellowship. He received the Best Paper Finalist from CVPR 2022 and the Best Student Paper Nomination from CVPR 2017. His team won the 1st place in the international challenges, including ActivityNet 2017, EPIC-Kitchens 2022, and Ego4D 2022 and 2023. He is on the Forbes 30 Under 30 Asia list.