This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Continual Segmentation with Disentangled Objectness Learning and Class Recognition

Yizheng Gong[1,2] Siyue Yu[1] Xiaoyang Wang[1,2,3] Jimin Xiao[1,*]
[1]Xi'an Jiaotong-Liverpool University [2]University of Liverpool [3]Metavisioncn
y.gong21@liverpool.ac.uk, siyue.yu@xjtlu.edu.cn, wangxy@liverpool.ac.uk, jimin.xiao@xjtlu.edu.cn

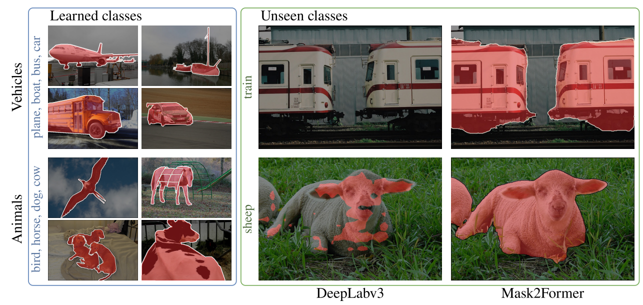
Abstract

Most continual segmentation methods tackle the problem as a per-pixel classification task. However, such a paradigm is very challenging, and we find query-based segmenters with built-in objectness have inherent advantages compared with per-pixel ones, as objectness has strong transfer ability and forgetting resistance. Based on these findings, we propose CoMasTRe by disentangling continual segmentation into two stages: forgetting-resistant continual objectness learning and well-researched continual classification. CoMasTRe uses a two-stage segmenter learning class-agnostic mask proposals at the first stage and leaving recognition to the second stage. During continual learning, a simple but effective distillation is adopted to strengthen objectness. To further mitigate the forgetting of old classes, we design a multi-label class distillation strategy suited for segmentation. We assess the effectiveness of CoMasTRe on PASCAL VOC and ADE20K. Extensive experiments show that our method outperforms per-pixel and query-based methods on both datasets. Code will be available at https://github.com/jordangong/CoMasTRe.

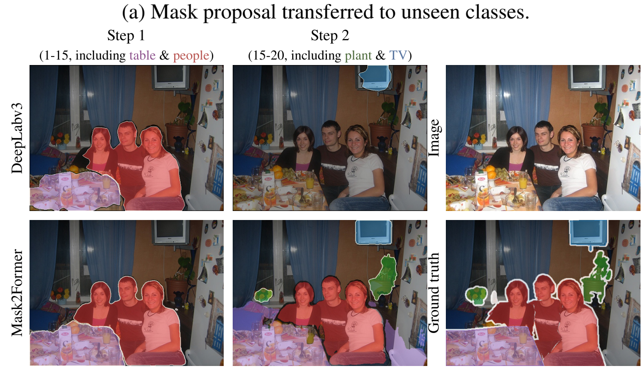
1. Introduction

Neural nets have dominated computer vision, from object recognition [18] to fine-grained classification [28, 29], segmentation [8, 48], and detection [59]. However, the success is only sound for single-shot learning, *i.e.*, training once to perform well with fully annotated datasets without considering the later learning process. *Continual learning* aims to mimic human learning by gradually obtaining knowledge in a sequential fashion [15, 17]. Particularly, continual learning shines in dense prediction tasks, such as semantic segmentation [3, 5, 12], since the annotations are laborious to obtain, and sometimes the learned samples are inaccessible, such as the nature of stream data in autonomous driving [34] or patient privacy in medical imaging [35].

*Corresponding author.



(a) Mask proposal transferred to unseen classes.



(b) Mask proposals after finetuning on new classes.

Figure 1. **Hidden properties inside query-based segmenters.** Objectness in query-based methods helps generalize mask proposals on unseen classes similar to learned classes (top). Additionally, because of the transfer ability of objectness, query-based methods are resistant to catastrophic forgetting of mask proposals (bottom).

However, machines tend to forget the early tasks if we train them to learn the new ones, also termed as *catastrophic forgetting* [16], which is the critical obstacle toward continual learning. Moreover, dense prediction tasks tackled as per-pixel classification or regression by analyzing patterns around each pixel are inherently difficult, which makes the continual learning of these tasks even more challenging. Could we mitigate forgetting in continual segmentation in a more feasible way? The answer lies in a new paradigm: *mask classification*, streamlining continual segmentation as learning without forgetting of class-agnostic binary masks and class recognition, via query-based segmenters [9, 10, 47, 55]. Different from conventional per-pixel-based segmenters [8, 42, 53], from our observation, query-based segmenters have strong objectness (the ability to find objects in images) built-in, which is beneficial to continual segmentation for two reasons. Firstly, as the background is not silenced when learning to propose masks, the objectness can be transferred to unseen classes. As illustrated in Fig. 1a, compared to DeepLabv3 [8], a per-pixel segmentation baseline, query-based Mask2Former [10] proposes substantially better masks on unseen classes if the model has learned similar classes before, *e.g*., learning vehicles, such as planes, boats, buses, and cars, facilitates the later learning of trains. Secondly, we observe that the objectness alleviates the forgetting of old class mask proposals (see Fig. 1b), as the model can still propose comparable masks on old classes, even finetuning on new classes.

Based on these findings, we propose **Co**ntinual Learning with **Mas**k-**T**hen-**Re**cognize **TR**ansformer decoder, termed as **CoMasTRe**, for continual segmentation. CoMasTRe disentangles the segmentation task to objectness learning and class recognition by learning to propose class-agnostic masks at the first stage and leaving recognition to the second stage. In continual settings, since CoMasTRe inherits the objectness of query-based methods, the learning of new mask proposals is considerably eased. This design also simplifies continual segmentation to forgetting-resistant continual objectness learning and a continual classification task. Specifically, CoMasTRe mitigates forgetting in two folds: (i) a simple but effective objectness distillation to reinforce old class objectness during long learning processes, and (ii) a multi-label class distillation strategy with task-specific classifiers to alleviate old class forgetting. For evaluation, we conduct extensive experiments on PASCAL VOC 2012 and ADE20K, showcasing that CoMasTRe reaches a new state-of-the-art on both datasets. In particular, we boost incremented class performance compared with previous best methods on VOC. Additionally, CoMasTRe significantly outperforms prior arts in all classes on ADE20K.

To sum up, the contributions of this paper are as follows:

- To leverage the properties of objectness, we propose **Co**ntinual learning with **Mas**k-**T**hen-**Re**cognize **TR**ansformer decoder, termed as CoMasTRe, to simply continual segmentation by decoupling the task into objectness learning and class recognition.
- To tackle the forgetting issue, we strengthen the objectness using a simple but effective distillation strategy in the first stage and preserve class knowledge in the second stage with task-specific classifiers and multi-label class distillation tailored to segmentation.
- Through extensive benchmarking on two datasets, our method outperforms prior query-based methods by up to 2.1% on ADE20K and surpasses per-pixel methods on new classes by up to 32.16% on PASCAL VOC.

2. Related Work

Query-based Image Segmentation. Query-based segmenters [9, 10, 21, 55] have unified image segmentation via *mask classification* and solve semantic, instance, and panoptic segmentation using the same framework. However, the *mask classification* paradigm was originated from query-based detectors, such as DETR [2] and its variants [24, 59, 66], where we train the model to predict proposals and class labels with learnable queries. MaskFormer [9] was the first to introduce *mask classification* in semantic segmentation, which was dominated by per-pixel classification paradigm via convolutional nets [8, 46, 52] or Transformers [42, 53]. Following MaskFormer, Mask2Former [10] introduced multiple improvements, such as masked attention for mask refinement and using multi-scale features for better small object performance. To speed up the convergence, kMaX-DeepLab [55] generated queries from clustered image features. Recently, OneFormer [21] provided a generalist solution by joint training of all segmentation tasks. However, all of these segmenters learn to predict to mask and class simultaneously and are not suitable for decoupled continual segmentation.

Continual Segmentation. Learning new knowledge by fintuning the model is a native way, but resulting *catastrophic forgetting* of old knowledge [16, 31, 39]. Continual learning aims to alleviate the forgetting issue by balancing the learning of new knowledge (*plasticity*) and the preserving of the old one (*stability*). While the research focus of continual learning is on classification [1, 6, 11, 22, 25, 38, 41, 50, 58], continual segmentation started to thrive recently [3, 5, 7, 12, 32, 36, 40, 51, 57, 60, 61, 65]. As pointed out in MiB [3], the background shift in continual segmentation aggravates the forgetting issue. Cermelli *et al*. [3] proposed unbiased distillation and classification to mitigate the background shift. PLOP [12] circumvented the background shift issue by feature distillation and pseudo-labeling. Based on PLOP, REMINDER [36] introduced class-weighted knowledge distillation to discriminate the new and old classes. With the same motivation, Incrementer [40] designed a class deconfusion strategy and achieved competitive performance via a Transformer-based segmenter. Recently, RL-Replay [65] set a strong replay-based baseline via reinforcement learning. Some works [4, 20] started to focus on more realistic scenarios by integrating weak supervision [56, 62] with continual learning. The methods above follow the *per-pixel* paradigm. Alhough CoMFormer [5] was the first to present *mask classification* in continual segmentation, it applied a standard framework with distillation and pseudo-labeling and failed to leverage the benefit of objectness. In contrast, CoMasTRe exploits objectness and decouples continual segmentation into forgetting-resistant objectness learning and class recognition.

Continual Dynamic Networks. In addition to distillation methods appearing in continual segmentation, dynamic networks with parameter expansion [13, 37, 41, 41, 45, 49, 50, 54, 64] also play a key role in continual learning. DER [54] is an early work on the dynamic structure by duplicating the backbone for each task to mitigate the forgetting issue. However, DER suffers from the explosion of parameters after a long learning process. FOSTER [45] relieved the problem through a model compression stage. MEMO [64] went further by decoupling the network and only expanding specialized layers. Recently, more works [13, 37, 41, 41, 49, 50] started to leverage the dynamic nature of Transformers. DyTox [13] proposed a dynamic token expansion mechanism by learning a task-specialized prompt at each step. The concurrent work, L2P [50], started a new fashion of continual learning: learning to prompt pretrained models and extracting task-specialized features. Thanks to our Transformer-based decoder, we adopt task queries and task-specific classifiers in our class decoder to further alleviate forgetting.

3. Method

3.1. Problem Definition

Image segmentation has been unified as a *mask classification* problem, *i.e.*, to propose class-agnostic masks and predict the corresponding class labels simultaneously. Formally, we define the dataset $\mathcal{D}$ containing pairs of image and target, and each sample pair contains an image $x \in \mathbb{R}^{C\times H\times W}$ and its target y, where C, H and W denote channel number, height and width of the image, respectively. Each target y is composed by M ground truth (GT) binary masks and class labels, denoted $y = \{(m_i^{\text{gt}}, c_i^{\text{gt}}) \mid m_i^{\text{gt}} \in \{0,1\}^{H\times W}, c_i^{\text{gt}} \in \mathcal{C}\}_{i=1}^{M}$, where $\mathcal{C}$ is the class label space. Unlike per-pixel classification methods, background class is excluded from annotation.

In terms of continual segmentation, the model sequentially learns to predict the masks of new classes but not to forget old ones. Formally, we divide the learning process to T tasks, and at each step $t = 1, 2, \dots, T$ we train the model to predict a unique set of classes $\mathcal{C}^t$, where $\bigcap_{t=i}^{T} \mathcal{C}^t = \varnothing$ and $\bigcup_{t=i}^{T} \mathcal{C}^t = \mathcal{C}$. Only the part of the annotations containing new classes $\mathcal{C}^t$ are provided at step t, while the model should be able to predict all learned classes $\mathcal{C}^{1:t}$.

3.2. CoMasTRe Architecture

Thanks to the properties of objectness (see Fig. 1) and widely studied continual classification, we argue that solving the continual segmentation problem with *mask classification* is inherently advantageous. Therefore, we propose CoMasTRe, which adopts Mask2Former [10] meta-architecture, keeping the backbone and the pixel decoder while adding two newly designed Transformer decoders to help disentangle objectness learning (stage 1) and class

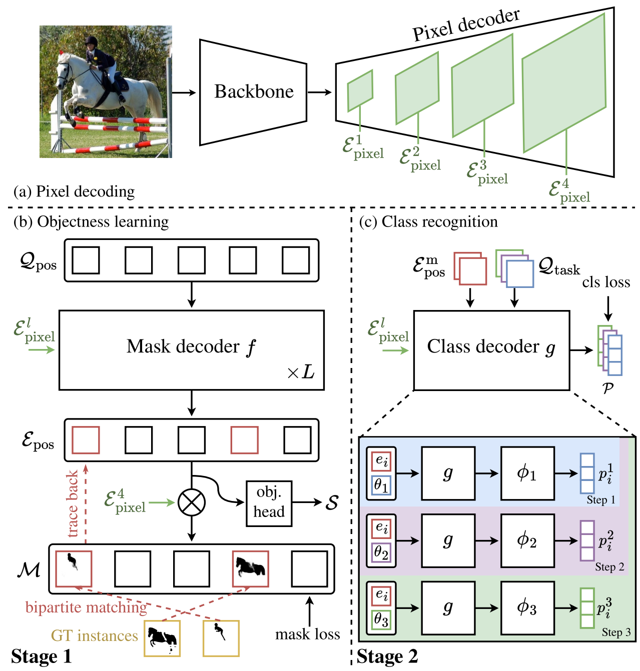


Figure 2. **CoMasTRe Architecture.** $\otimes$ denotes the dot product between positional embeddings $\mathcal{E}_{\text{pos}}$ and pixel embeddings $\mathcal{E}^4_{\text{pixel}}$. CoMasTRe uses a two-stage image segmenter including three components: (a) a backbone and a pixel decoder producing pixel embedding, (b) a mask decoder f with learnable positional queries $\mathcal{Q}_{\text{pos}}$ for objectness learning, and (c) a class decoder g with a set of task queries $\mathcal{Q}_{\text{task}}$ for class recognition.

recognition (stage 2), as shown in Fig. 2. The overall training process of our CoMasTRe can be summarized as:

1. When an image comes to CoMasTRe, the backbone, and the pixel decoder first encode and then decode the image to pixel embeddings. We take out the pixel embeddings from all the layers of the pixel decoder for the following process as $\{\mathcal{E}^l_{\text{pixel}}\}_{l=1}^4$, where l denotes different layers and there are 4 layers in total (Fig. 2 (a)).
2. Next, we randomly initialize learnable positional queries $\mathcal{Q}_{\text{pos}}$ and input them to the objectness learning stage (stage 1), to extract positional embeddings $\mathcal{E}_{\text{pos}}$ through the mask decoder f for predicting class-agnostic mask proposals $\mathcal{M}$ and objectness scores $\mathcal{S}$. The training involves bipartite matching with ground truth (Fig. 2 (b)).
3. After that, matched positional embeddings $\mathcal{E}^{\text{m}}_{\text{pos}}$ from stage 1 with pixel embeddings are input to class decoder g for recognition (stage 2). To reduce task interference during continual learning, task queries $\mathcal{Q}_{\text{task}}$ and task-specific classifiers $\phi_1, \dots, \phi_t$ are learned to specialize the class knowledge of each task. (Fig. 2 (c)).
4. Finally, we obtain segmentation results by combining the mask proposals and objectness scores from stage 1 with the class prediction from stage 2.

The details of the two stages will be illustrated in Sec. 3.2.1 and Sec. 3.2.2.

3.2.1 Stage 1: Objectness Learning

As illustrated in Fig. 2 (b), our objectness learning mainly relies on a mask decoder. The mask decoder f is composed of L blocks of Transformer layers [44], taking N learnable positional queries $\mathcal{Q}_{\text{pos}} = \{q_1, \ldots, q_N\} \in \mathbb{R}^{N \times d}$ and intermediate pixel embeddings $\{\mathcal{E}^l_{\text{pixel}}\}^3_{l=1}$ as input. Then, it outputs the positional embeddings $\mathcal{E}_{\text{pos}} = \{e_1, \ldots, e_N\} \in \mathbb{R}^{N \times d}$ for mask proposals, objectness scores, and stage 2 recognition. The mask proposals $\mathcal{M} = \{m_1, \ldots, m_N\} \in [0,1]^{N \times H \times W}$ is computed as $\mathcal{M} = \text{sigmoid}(\text{Upsample}(\text{MLP}(\mathcal{E}_{\text{pos}}) \cdot \mathcal{E}^4_{\text{pixel}}))$, where the $\text{MLP}(\cdot)$ acts as a non-linear transformation function and $\text{Upsample}(\cdot)$ interpolates the logits to image size. Note here each positional embedding produces a mask proposal. At the same time, $\mathcal{E}_{\text{pos}}$ is input to our objectness head, in which a binary classifier is designed to output the objectness score $\mathcal{S} = \{s_1, \ldots, s_N\} \in [0,1]^N$ to indicate whether the mask proposals containing objects or not. During training, we first perform bipartite matching considering the cost between N mask proposals $\{m_i\}^N_{i=1}$ and M ground truth masks $\{m^{\text{gt}}_j\}^M_{j=1}$. Here, we assume $N \gg M$ and pad the ground truth with "no object" $\varnothing$ to ensure a one-to-one matching between predictions and ground truth. After solving the matching via the Hungarian algorithm [23], we get the optimal permutation of N elements, denoted as σ. Then, we optimize this stage using ground truth masks:

$$\begin{aligned}\mathcal{L}_{\text{obj}} = \sum_{j=1}^{N} \big[-\mathbb{1}_{m^{\text{gt}}_j \neq \varnothing} \log s_{\sigma_j} - \mathbb{1}_{m^{\text{gt}}_j = \varnothing} \log(1 - s_{\sigma_j}) \\ + \mathbb{1}_{m^{\text{gt}}_j \neq \varnothing} \mathcal{L}_{\text{mask}}(m_{\sigma_j}, m^{\text{gt}}_j) \big],\end{aligned} \tag{1}$$

where $\mathcal{L}_{\text{mask}}$ is the sum of binary cross-entropy loss and Dice loss [43]. Based on the matching results, M matched positional embeddings $\mathcal{E}^{\text{m}}_{\text{pos}} = \{e_j \mid e_j \in \mathcal{E}_{\text{pos}}, m^{\text{gt}}_j \neq \varnothing\}$ are traced back and fed forward to stage 2 class decoder. Here, we supervise the objectness scores by encouraging the matched ones and suppressing the unmatched ones, otherwise, the model tends to produce high objectness scores for all the proposals. The remaining unmatched embeddings $\mathcal{E}^{\text{u}}_{\text{pos}} = \mathcal{E}_{\text{pos}} \setminus \mathcal{E}^{\text{m}}_{\text{pos}}$ are reserved for distillation to alleviate forgetting (see Sec. 3.3).

3.2.2 Stage 2: Class Recognition

Our stage 2 aims to recognize the object inside the mask proposals. Thus, we use matched positional embeddings along with pixel embeddings as input for class recognition. Besides, inspired by DyTox [13], we integrate task queries with matched positional embeddings as input to task-specific classifiers in the class decoding process, which facilitates continual learning by reducing task interference, resulting in more specialized classifiers.

Since continual segmentation contains T tasks, to illustrate our class recognition stage, we assume the current learning step is t. We denote task queries $\mathcal{Q}_{\text{task}} = \{\theta_1, \ldots, \theta_t\}$. As shown in Fig. 2 (c), for each positional embedding $e \in \mathcal{E}^{\text{m}}_{\text{pos}}$ and each task query $\theta \in \mathcal{Q}_{\text{task}}$, we take (e, θ) as a pair and feed it with pixel embedding to the class decoder g (a Transformer block shared across tasks), obtaining the localized task embedding k. For each positional embedding, we obtain its task embeddings $\mathcal{E}_{\text{task}} = \{k_1, \ldots, k_t\}$. Then, t task-specific classifiers $\Phi^t = \{\phi_1, \ldots, \phi_t\}$ are applied to $\mathcal{E}_{\text{task}}$, where each classifier $\phi_j \in \Phi^t$ is specialized to predict the classes in this task $\mathcal{C}^j$, *i.e.*, $\phi_j : \mathbb{R}^d \mapsto \mathcal{C}^j$. In this way, we obtain the class probability of each proposal as $p = \text{sigmoid}([\phi_1(k_1), \ldots, \phi_t(k_t)])$, as each positional embedding corresponds to a mask proposal. By feeding all the positional embeddings, we get the class probability of all proposals $\mathcal{P} = \{p_i\}^M_{i=1}$. We supervise this stage with classification loss using ground truth class labels:

$$\begin{aligned}\mathcal{L}_{\text{cls}} = -\frac{1}{M} \sum_{i=1}^{M} \sum_{c \in \mathcal{C}^t} \big[\mathbb{1}_{c = c^{\text{gt}}_i} (1 - p_{i,c})^\gamma \log p_{i,c} \\ + \mathbb{1}_{c \neq c^{\text{gt}}_i} p^\gamma_{i,c} \log(1 - p_{i,c}) \big],\end{aligned} \tag{2}$$

where $p_{i,c}$ is the predicted probability of class c for the i-th proposal, γ is a hyperparameter. Here, we use focal loss [27] instead of cross-entropy loss for better calibration [33], which smooths the distillation process during continual learning. The overall segmentation loss is the sum of objectness loss and classification loss, *i.e.*, $\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{cls}}$. We jointly train two stages by minimizing $\mathcal{L}_{\text{seg}}$. During inference, we set an objectness threshold α at stage 1 to filter low confident prediction and feed high objectness embedding to stage 2 for mask recognition.

3.3. Learning without Forgetting with CoMasTRe

Learning new classes by naively finetuning the model will cause catastrophic forgetting. To alleviate this issue, our CoMasTRe separately considers the distillation of both stages. In this way, predictions on base classes can also be preserved when learning new classes.

3.3.1 Objectness Distillation

As shown in Fig. 3 (a), we first train the model with bipartite matching and get matched positional embeddings and unmatched ones. However, the objectness scores of unmatched ones quickly diminish as the ground truth does not contain masks from old classes, resulting in incorrect objectness scores on old classes. Thus, we introduce objectness score distillation with $\mathcal{L}_{\text{os-kd}}$ to preserve objectness scores learned in previous steps. Moreover, in spite of the robust nature of objectness, could we further improve the objectness during continual learning? To answer this question, we propose another two distillation losses $\mathcal{L}_{\text{mask-kd}}$ and $\mathcal{L}_{\text{pe-kd}}$ for mask proposals and position embeddings, respectively (see Fig. 3 (b)). Following previous methods [3, 25], we distill the outputs from the last step model to mitigate the objectness forgetting on old classes.

Formally, we denote unmatched mask proposals as $\mathcal{M}^{\text{u}} = \{m_1^{\text{u}}, \dots, m_{N-M}^{\text{u}}\}$ and objectness scores as $\mathcal{S}^{\text{u}} = \{s_1^{\text{u}}, \dots, s_{N-M}^{\text{u}}\}$, which are the corresponding outputs from unmatched positional embeddings $\mathcal{E}_{\text{pos}}^{\text{u}} = \{e_i^{\text{u}}, \dots, e_{N-M}^{\text{u}}\}$. Additionally, the distillation process requires the corresponding outputs from the last step model, including mask proposals $\tilde{\mathcal{M}}^{\text{u}} = \{\tilde{m}_1^{\text{u}}, \dots, \tilde{m}_{N-M}^{\text{u}}\}$, objectness scores $\tilde{\mathcal{S}}^{\text{u}} = \{\tilde{s}_1^{\text{u}}, \dots, \tilde{s}_{N-M}^{\text{u}}\}$, and positional embeddings $\tilde{\mathcal{E}}_{\text{pos}}^{\text{u}} = \{\tilde{e}_1^{\text{u}}, \dots, \tilde{e}_{N-M}^{\text{u}}\}$. We freeze the parameters of the last step model during distillation.

Distill objectness scores. We use vanilla knowledge distillation loss [19] to mitigate the diminishing of objectness scores by minimizing Kullback-Leibler (KL) divergence between the current unmatched objectness score $\mathcal{S}^{\text{u}}$ and the last step ones $\tilde{\mathcal{S}}^{\text{u}}$,

$$\mathcal{L}_{\text{os-kd}} = -\frac{1}{|\mathcal{S}^{\text{u}}|}\sum_{i=1}^{|\mathcal{S}^{\text{u}}|}\left[\tilde{s}_i^{\text{u}} \log \frac{s_i^{\text{u}}}{\tilde{s}_i^{\text{u}}} + (1-\tilde{s}_i^{\text{u}}) \log \frac{1-s_i^{\text{u}}}{1-\tilde{s}_i^{\text{u}}}\right]. \tag{3}$$

As objectness scores play a key role during inference, this distillation is designed to maintain the objectness scores of previously learned mask proposals.

Distill mask proposals. To preserve the knowledge of the mask proposals but also focus on proposals with high objectness scores, we reweight mask distillation with objectness scores. By distilling masks with low objectness scores, we relax the case when objectness scores fail to indicate the objectness, as we observe some mask proposals generalize to unseen classes, but their objectness scores are low. The mask distillation loss is defined as

$$\mathcal{L}_{\text{mask-kd}} = \sum_{i=1}^{|\mathcal{M}^{\text{u}}|} \frac{\omega_i}{\sum_{i=1}^{|\mathcal{M}^{\text{u}}|}\omega_i} \mathcal{L}_{\text{mask}}(m_i^{\text{u}}, \tilde{m}_i^{\text{u}}), \tag{4}$$

where $\omega_i = (\tilde{s}_i^{\text{u}})^{\beta}$, and β is a reweight strength hyperparameter.

Distill positional embeddings. Similar to mask distillation, we reweight position distillation when enforcing the unmatched positional embeddings $\mathcal{E}_{\text{pos}}^{\text{u}}$ to be similar to the last step ones $\tilde{\mathcal{E}}_{\text{pos}}^{\text{u}}$. Here, we encourage high cosine similarity between the current step ones and the last step ones, which assists in preserving positional information for later multi-label class distillation,

$$\mathcal{L}_{\text{pe-kd}} = \sum_{i=1}^{|\mathcal{E}_{\text{pos}}^{\text{u}}|} \frac{\omega_i}{\sum_{i=1}^{|\mathcal{E}_{\text{pos}}^{\text{u}}|}\omega_i} \left(1-\cos(e_i^{\text{u}}, \tilde{e}_i^{\text{u}})\right), \tag{5}$$

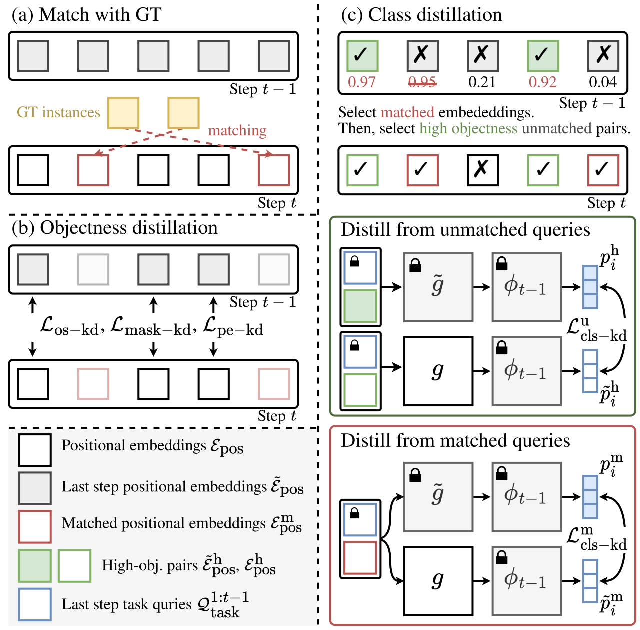


Figure 3. **Learning without forgetting with CoMasTRe.** To tackle catastrophic forgetting, CoMasTRe separated the distillation process into two stages, including objectness distillation and class distillation. We perform bipartite matching first, as in (a). Then, we distill the knowledge of objectness for remaining embeddings, as in (b). Finally, we select positional embeddings for the class distillation if they match with ground truth or have high objectness scores at the last step. The class knowledge is distilled from both matched and unmatched positional embeddings with a class decoder and task-specific classifiers, as in (c).

where $\cos(\cdot)$ means cosine similarity.

Finally, we distill the objectness by minimizing $\mathcal{L}_{\text{obj-kd}} = \mathcal{L}_{\text{os-kd}} + \mathcal{L}_{\text{mask-kd}} + \mathcal{L}_{\text{pe-kd}}$.

3.3.2 Class Distillation

As the segmentation is decoupled to a recognition task at the second stage, the continual learning of the second stage is essentially continual multi-label classification. We discuss class distillation in two scenarios: (i) distilling the class prediction from the last step high-objectness embeddings unmatched with GT, and (ii) distilling the class knowledge from matched embeddings (see in Fig. 3 (c)). During class distillation, we freeze the parameters of the last step model.

Distill from unmatched queries. This scenario is similar to the pseudo-labeling of the old classes but in a soft way, minimizing the KL divergence between the class probability of the current step and the last step. Given the last step unmatched embeddings $\tilde{\mathcal{E}}_{\text{pos}}^{\text{u}}$, we select n embeddings with high objectness score $\tilde{\mathcal{E}}_{\text{pos}}^{\text{h}} = \{\tilde{e}_i^{\text{u}} \mid \tilde{s}_i^{\text{u}} > \alpha\} \in \mathbb{R}^{n \times d}$, where α is high objectness threshold. The corresponding current step embeddings $\mathcal{E}_{\text{pos}}^{\text{h}}$ are also selected. After class decoding, we obtain the last step class logits

$\{\tilde{z}_i^{\mathrm{h}}\}_{i=1}^n$ and the current ones $\{z_i^{\mathrm{h}}\}_{i=1}^n$. The knowledge distillation loss is enforced on the probability of old classes $\mathcal{C}^{1:t-1}$, *i.e.*, $\mathcal{L}_{\mathrm{cls-kd}}^{\mathrm{u}} = \frac{1}{n}\sum_{i=1}^n D_{\mathrm{KL}}(p_i^{\mathrm{h}} \| \tilde{p}_i^{\mathrm{h}})$, where $p_i^{\mathrm{h}} = \mathrm{softmax}(z_i^{\mathrm{h}})$ and $\tilde{p}_i^{\mathrm{h}} = \mathrm{softmax}(\tilde{z}_i^{\mathrm{h}})$ are the current step old class probability and the last step one decoded from i-th positional embedding. $D_{\mathrm{KL}}(\cdot)$ denotes KL divergence.

Distill from matched queries. For new samples, we consider its last step model prediction as the prior knowledge to mitigate forgetting. To extract the knowledge, we reuse the currently matched embeddings $\mathcal{E}_{\mathrm{pos}}^{\mathrm{m}}$ and get the old class probability activated by softmax $\{p_i^{\mathrm{m}}\}_{i=1}^M$. Additionally, we decode $\mathcal{E}_{\mathrm{pos}}^{\mathrm{m}}$ using the last step model, obtaining the last step probability $\{\tilde{p}_i^{\mathrm{m}}\}_{i=1}^M$ as the target distribution. During training, we minimize matched class distillation loss $\mathcal{L}_{\mathrm{cls-kd}}^{\mathrm{m}} = \frac{1}{M}\sum_{i=1}^M D_{\mathrm{KL}}(p_i^{\mathrm{m}} \| \tilde{p}_i^{\mathrm{m}})$.

In total, we preserve class knowledge by minimizing $\mathcal{L}_{\mathrm{cls-kd}} = \mathcal{L}_{\mathrm{cls-kd}}^{\mathrm{u}} + \mathcal{L}_{\mathrm{cls-kd}}^{\mathrm{m}}$. Besides, we follow prior classification works [13, 54] and use an auxiliary loss $\mathcal{L}_{\mathrm{cls-aux}}$ with a similar purpose to previous continual segmentation methods [36, 40]. It encourages more discriminative classification of similar classes by learning to predict a merged old class and new classes (1 + $|\mathcal{C}^t|$ classes in total) at each step. Together with segmentation loss, the loss of CoMasTRe is $\mathcal{L} = \mathcal{L}_{\mathrm{seg}} + \mathcal{L}_{\mathrm{kd}} + \mathcal{L}_{\mathrm{cls-aux}}$, where $\mathcal{L}_{\mathrm{kd}} = \mathcal{L}_{\mathrm{obj-kd}} + \mathcal{L}_{\mathrm{cls-kd}}$.

4. Experiments

4.1. Setup

Datasets. Following standard continual segmentation baselines [3, 12, 36], we evaluate semantic segmentation performance of CoMasTRe on PASCAL VOC 2012 [14] and ADE20K [63]. PASCAL VOC 2012 is a relatively small dataset with 20 object classes and background class, containing 10,582 samples for training and 1,449 for validation. ADE20K is a large-scale semantic segmentation dataset with 150 annotated classes, containing 20,210 and 2,000 samples for training and validation.

Continual Segmentation Protocols. Continual segmentation protocols include *sequential*, *disjoint*, and *overlapped*, where all of them split the segmentation into T steps, and each step aims to learn a unique set of classes. The *sequential* is the easiest, as the training set contains the ground truth of each pixel. In contrast, the *disjoint* annotates past classes as background and excludes any training pixel belonging to future classes. The *overlapped* is the most challenging and realistic one, where the dataset only contains the ground truth of current classes, but both past and future classes are annotated as background. Following previous works [5, 12, 36], we adopt the *overlapped* in benchmarks.

We evaluate CoMasTRe in 6 settings, 3 on PASCAL VOC and 3 on ADE20K. Here, we use the B-I notation in the paper, where B and I denote the number of base classes and incremented classes per step. Then, the number of total steps $T = 1 + (|\mathcal{C}| - B)/I$. For example, *15-1* on VOC means we start the learning process at 15 base classes, then continually train the model for 5 steps, one new class per step. In this way, the 3 settings on PASCAL VOC are (i) *19-1*, 2-step learning starting with 19 classes and following 1 class, (ii) *15-5*, 2-step learning starting with 15 classes and following 5 classes, and (iii) *15-1*, consisting 6 learning steps with 15 classes at first and following 5 steps with one class each step. Similarly, the settings on ADE20K are *100-50*, *100-10*, and *100-5*. Note that the longer the learning process, the harder it is to tackle forgetting.

Metric. We evaluate semantic segmentation performance using mean intersection over union (mIoU). In continual learning settings, the mIoU of base classes $\mathcal{C}^1$ and incremented classes $\mathcal{C}^{2:T}$ are also measured. The base class performance measures the *stability* of the model, while the incremented one reflects the *plasticity*. Additionally, following [5, 12, 36], we report the average mIoU of all learning steps as *avg* to evaluate the entire continual learning process. We denote *joint* as the joint training of all classes.

Implementation details. Following the previous continual segmentation benchmarks [3, 5, 12], we use ImageNet-pretrained ResNet101 [18] as the backbone. The pixel decoder in CoMasTRe is identical to the Mask2Former [10] one, the mask decoder consists of $L = 9$ Transformer decoder blocks, and the class decoder consists of one block. We set the number of positional queries N to 20 and 100 for PASCAL VOC and ADE20K, respectively. During training, we follow the Mask2Former [10] hyperparameters. We use AdamW [30] optimizer with an initial learning rate of 1×10^{-4} and weight decay of 0.05 and apply a polynomial learning rate schedule. We train the model with a batch size of 16 for 20K iterations on PASCAL VOC and 160K iterations on ADE20K at the first learning step. In later steps, we half the learning rate to 5×10^{-5} and train for 1,000 iterations per class on VOC and 500 iterations per class on ADE20K. The images are augmented with random rescaling, flipping, and color jittering then cropped to 512×512 resolution. We set reweight parameters γ and β to 2.0. To evaluate the IoU of background during continual learning, we use the panoptic inference from Mask2Former [10]. During inference, we set the high objectness threshold α as 0.8. Following CoMFormer [5], we use single-scale inference, and no replay is involved throughout training.

4.2. Quantitative Evaluation

We compare CoMasTRe with state-of-the-art continual segmentation methods on PASCAL VOC 2012 and ADE20K. Qualitative results are available in supplementary materials.

PASCAL VOC 2012. We compare our method with previous state-of-the-art methods on the 3 settings on PASCAL VOC: *19-1*, *15-5*, and *15-1* in Tab. 1. In general, CoMasTRe

Table 1. Comparison with previous best methods on PASCAL VOC 2012 in mIoU (%). The highest and the second highest results are marked in **bold** and underline. $*$ means results from our re-implementation.

Paradigm	Method	**19-1** (2 tasks)				**15-5** (2 tasks)				**15-1** (6 tasks)			
		1-19	*20*	*all*	*avg*	*1-15*	*16-20*	*all*	*avg*	*1-15*	*16-20*	*all*	*avg*
Per-Pixel	MiB [3]	71.43	23.59	69.15	73.28	76.37	49.97	70.08	75.12	34.22	13.50	29.29	54.19
	SDR [32]	68.52	23.29	66.37	71.48	75.21	46.72	68.64	74.32	43.08	19.31	37.42	54.52
	PLOP [12]	75.35	37.35	73.54	75.47	75.73	51.71	70.09	75.19	65.12	21.11	54.64	67.21
	REMINDER [36]	**76.48**	32.34	74.38	76.22	76.11	50.74	70.07	75.36	68.30	27.23	58.52	68.27
	RCIL [57]	—	—	—	—	78.80	**52.00**	72.40	—	**70.60**	23.70	59.40	—
	Joint	77.45	77.94	77.39	—	78.88	72.63	77.39	—	78.88	72.63	77.39	—
Query	CoMFormer* [5]	75.35	24.06	72.91	75.46	74.68	48.47	68.44	72.97	48.97	23.28	48.18	64.16
	Joint	77.09	71.39	76.82	—	78.60	71.12	76.82	—	78.60	71.12	76.82	—
	CoMasTRe (ours)	75.13	**69.51**	**74.86**	**76.66**	**79.73**	51.93	**73.11**	**75.97**	69.77	**43.62**	**63.54**	**70.63**
	Joint	78.57	68.32	78.08	—	80.83	69.28	78.08	—	80.83	69.28	78.08	—

Table 2. Comparison with previous methods on ADE20K in mIoU (%). The 1st and 2nd highest results are marked in **bold** and underline.

Paradigm	Method	**100-50** (2 tasks)				**100-10** (6 tasks)				**100-5** (11 tasks)			
		1-100	*101-150*	*all*	*avg*	*1-100*	*101-150*	*all*	*avg*	*1-100*	*101-150*	*all*	*avg*
Per-Pixel	MiB [3]	40.50	17.20	32.80	37.30	38.30	11.30	29.20	35.10	36.00	5.70	26.00	32.70
	SDR [32]	40.52	17.17	32.79	37.31	37.26	12.13	28.94	34.48	33.02	10.63	25.61	33.07
	PLOP [12]	41.87	14.89	32.94	37.39	40.48	13.61	31.59	36.64	35.72	12.18	27.93	35.10
	REMINDER [36]	41.55	19.16	34.14	38.43	38.96	**21.28**	33.11	37.47	36.06	**16.38**	29.54	36.49
	RCIL [57]	42.30	18.80	34.50	—	39.30	17.60	32.10	—	38.50	11.50	29.60	—
	Joint	44.34	28.21	39.00	—	44.34	28.21	39.00	—	44.34	28.21	39.00	—
Query	CoMFormer [5]	44.70	**26.20**	38.40	41.20	40.60	15.60	32.30	37.40	39.50	13.60	30.90	36.50
	Joint	46.90	35.60	43.10	—	46.90	35.60	43.10	—	46.90	35.60	43.10	—
	CoMasTRe (ours)	**45.73**	26.02	**39.20**	**41.62**	**42.32**	18.42	**34.41**	**38.41**	**40.82**	15.83	**32.55**	**38.64**
	Joint	48.48	36.11	44.36	—	48.48	36.11	44.36	—	48.48	36.11	44.36	—

significantly outperforms query-based CoMFormer [5] and surpasses per-pixel baselines on incremented classes by a large margin. On *19-1*, CoMasTRe shows off its strong plasticity in learning new classes, with 32.16 percent points (*p.p*) improvement while maintaining the knowledge of old classes. On *15-5*, our model is on par with RCIL [57] on new classes while exceptionally great at maintaining old class performance, even surpassing the *joint* upper bound of per-pixel baselines by 0.85 *p.p*. *15-1* is much more difficult and involves 6 learning steps. However, CoMasTRe still beats RCIL with 4.14 *p.p* uplift on *all* classes. Overall, we strikes a balance between *stability* and *plasticity*.

ADE20K. As shown in Tab. 2, we report the results on ADE20K in *100-50*, *100-10*, and *100-5*. In general, CoMasTRe outperforms previous state-of-the-art CoMFormer [5] in each setting by 0.8 *p.p*, 2.11 *p.p*, and 1.65 *p.p* on *all* classes, respectively. In *100-50*, our method shows competitive performance with CoMFormer on incremented classes but with 0.8 *p.p* gain on *all* classes thanks to its higher upper bound. For longer learning processes, CoMasTRe reaches a comparable performance with REMINDER [36] on new classes but significantly boosts base class mIoU (3.36 *p.p* in *100-10* and 4.76 *p.p* in *100-5*). Compared with CoMFormer, our CoMasTRe also learns better new classes with 2.82 *p.p* improvement in *100-10* and 2.23 *p.p* in *100-5* while preserving better old class knowledge, with 1.72 *p.p* and 1.32 *p.p* gains, respectively.

4.3. Ablation Studies

Joint training results. To resolve scalability concerns on mask-only matching, we demonstrate the *joint* performance of CoMasTRe, *i.e.*, single-shot training on all classes. We compare the segmentation performance in mIoU on PASCAL VOC and ADE20k using ResNet50 (R50) and ResNet101 (R101) backbones in Tab. 3. The results show that CoMasTRe achieves comparable performance with Mask2Former and outperforms the previous continual model CoMFormer by $\sim$1.2 *p.p* on ADE20K.

Objectness transfer ability analysis. To study the transfer ability of objectness, we first pretrain a stage 1 mask decoder on modified COCO 2017 [26], where instances of the classes appearing in PASCAL VOC are removed to prevent information leaks. Then, we transfer the stage 1 and continually train it on 15 base classes of VOC with objectness distillation. Finally, we follow standard procedures by training 1 class per step. As shown in Tab. 4, compared with the model trained on VOC only, we get 1.34 *p.p* performance uplift on *all* metric in PASCAL VOC *15-1* setting, indicating the strong transfer ability of objectness.

Effectiveness of objectness distillation. We ablate object-

Table 3. Semantic segmentation results mIoU in *joint* setting. Following continual settings, we use panoptic-style inference. The 1st and 2nd highest results are marked in **bold** and underline.

Method	Backbone	Datasets VOC	ADE20k
Mask2Former [10]	R50	**79.73**	<u>43.28</u>
	R101	**80.04**	<u>43.51</u>
CoMFormer [5]	R50	76.01	42.58
	R101	76.82	43.10
CoMasTRe (ours)	R50	<u>77.84</u>	**43.97**
	R101	<u>78.08</u>	**44.36**

Table 4. Objectness transfer ability results in PASCAL VOC *15-1* setting. For transfer learning, we continually train on VOC using a stage 1 mask decoder pretrained on COCO (w/o PASCAL classes). Otherwise, the model is trained on VOC only without pretraining.

Transfer	*1-15*	*16-20*	*all*	*avg*
✓	**70.72**	**46.18**	**64.88**	**71.34**
✗	69.77	43.62	63.54	70.63

Table 5. Objectness distillation results in PASCAL VOC *15-1* setting, where $\mathcal{L}_{\text{mask-kd}}$ for mask distillation, $\mathcal{L}_{\text{os-kd}}$ for objectness score distillation, $\mathcal{L}_{\text{pe-kd}}$ for position distillation.

$\mathcal{L}_{\text{mask-kd}}$	$\mathcal{L}_{\text{os-kd}}$	$\mathcal{L}_{\text{pe-kd}}$	*1-15*	*16-20*	*all*	*avg*
✓			13.14	35.42	18.44	47.43
	✓		63.75	37.97	57.61	62.72
✓	✓		66.23	38.57	59.64	64.61
✓	✓	✓	**69.77**	**43.62**	**63.54**	**70.63**

ness distillation components with optimal configuration on other parts. The ablation is conducted in four cases: (1) $\mathcal{L}_{\text{mask-kd}}$ for mask only, (2) $\mathcal{L}_{\text{os-kd}}$ for objectness score only, (3) $\mathcal{L}_{\text{mask-kd}}+\mathcal{L}_{\text{os-kd}}$ for mask and objectness score, and (4) $\mathcal{L}_{\text{mask-kd}} + \mathcal{L}_{\text{os-kd}} + \mathcal{L}_{\text{pe-kd}}$ for mask, objectness score, and position. As shown in Tab. 5, we report the performance in PASCAL VOC *15-1* setting. By comparing case 1 and 3, without $\mathcal{L}_{\text{os-kd}}$, base class mIoU degrades significantly as the base objectness score diminishes. By comparing case 2 and 3, we observe slight forgetting of mask proposals without $\mathcal{L}_{\text{mask-kd}}$ ($\sim$2 *p.p* drop on *all* metric), showing the forgetting robustness of mask proposals. The position distillation aims for better alignment of localized information and leads to better classification results, with $\sim$4 *p.p all* performance gain when comparing case 3 and 4.

Effectiveness of class distillation. We perform ablation in VOC *15-1* setting by analyzing the effectiveness of $\mathcal{L}^{\text{u}}_{\text{cls-kd}}$ and $\mathcal{L}^{\text{m}}_{\text{cls-kd}}$ in Sec. 3.3.2. The results in Tab. 6 indicate $\mathcal{L}^{\text{u}}_{\text{cls-kd}}$ playing a pivotal role in alleviating forgetting, coinciding with pseudo-labeling in previous methods [5, 12]. With $\mathcal{L}^{\text{m}}_{\text{cls-kd}}$, the forgetting issue is mitigated even further.

Effectiveness of stage 2 other components. As shown in Tab. 7, under optimal distillation configurations, we investigate the contribution of stage 2 other components,

Table 6. Ablation results on class distillation in VOC *15-1* setting.

$\mathcal{L}^{\text{m}}_{\text{cls-kd}}$	$\mathcal{L}^{\text{u}}_{\text{cls-kd}}$	*1-15*	*16-20*	*all*	*avg*
✓		34.63	42.36	36.47	53.92
	✓	65.24	39.53	59.12	64.23
✓	✓	**69.77**	**43.62**	**63.54**	**70.63**

Table 7. More ablation results on stage 2 components in PASCAL VOC *15-1* setting, where "TQ" denotes task queries, "Aux" denotes auxiliary loss $\mathcal{L}_{\text{cls-aux}}$, and "Focal" denotes focal loss $\mathcal{L}_{\text{cls}}$.

TQ	Aux	Focal	*1-15*	*16-20*	*all*	*avg*
			59.30	25.96	51.36	59.44
✓			63.66	29.32	55.48	62.45
	✓		65.21	35.53	58.14	63.93
		✓	66.43	40.20	60.18	64.72
✓	✓		67.91	38.41	60.89	65.06
✓	✓	✓	**69.77**	**43.62**	**63.54**	**70.63**

including task queries (TQ), focal loss $\mathcal{L}_{\text{cls}}$ (Focal) in Sec. 3.2.2, and auxiliary loss $\mathcal{L}_{\text{cls-aux}}$ (Aux) in Sec. 3.3.2. Note that when not using task queries, one query is used across all tasks; when not using focal loss, cross-entropy is applied. We first ablate the improvements with each improvement only. First, with task queries, we get $\sim$4 *p.p* improvement on *all* metric as they reduce the learning interference between tasks. Second, the auxiliary loss assists in learning without forgetting new classes similar to old classes by increasing $\sim$7 *p.p* of *all* metric. Third, when using the focal loss to smooth the class probability distribution, it facilitates the distillation process and results in $\sim$9 *p.p* mIoU gain. With all the improvements, we boost the final performance by $\sim$12 *p.p*.

5. Conclusion

In this paper, we present CoMasTRe, a continual segmentation framework by disentangling the challenging segmentation problem to objectness learning and class recognition. To leverage the properties of objectness, we propose a two-stage query-based segmenter and distill objectness and classification knowledge separately to alleviate forgetting. Extensive experiments show that our method achieves considerably better performance compared with state-of-the-art methods on PASCAL VOC 2012 and ADE20K. For future works, CoMasTRe provides a decoupled way to tackle continual semantic segmentation and could be extended to continual panoptic and instance segmentation.

Acknowledgments. This work was supported by the National Key R&D Program of China (No. 2022YFE0200300), the National Natural Science Foundation of China (No. 61972323, 62331003), Suzhou Basic Research Program (SYG202316) and XJTLU REF-22-01-010, XJTLU AI University Research Centre, Jiangsu Province Engineering Research Centre of Data Science and Cognitive Computation at XJTLU and SIP AI innovation platform (YZCXPT2022103).

References

[1] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning: A Strong, Simple Baseline. In *NeurIPS*, 2020. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *ECCV*, 2020. 2

[3] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the Background for Incremental Learning in Semantic Segmentation. In *CVPR*, 2020. 1, 2, 5, 6, 7

[4] Fabio Cermelli, Dario Fontanel, Antonio Tavera, Marco Ciccone, and Barbara Caputo. Incremental Learning in Semantic Segmentation from Image Labels. In *CVPR*, 2022. 2

[5] Fabio Cermelli, Matthieu Cord, and Arthur Douillard. CoMFormer: Continual Learning in Semantic and Panoptic Segmentation. In *CVPR*, 2023. 1, 2, 6, 7, 8

[6] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2L: Contrastive Continual Learning. In *ICCV*, 2021. 2

[7] Sungmin Cha, Beomyoung Kim, YoungJoon Yoo, and Taesup Moon. SSUL: Semantic Segmentation with Unknown Label for Exemplar-based Class-Incremental Learning. In *NeurIPS*, 2021. 2

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. In *NeurIPS*, 2021. 2

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-Attention Mask Transformer for Universal Image Segmentation. In *CVPR*, 2022. 2, 3, 6, 8

[11] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *ECCV*, 2020. 2

[12] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. PLOP: Learning without Forgetting for Continual Semantic Segmentation. In *CVPR*, 2021. 1, 2, 6, 7, 8

[13] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion. In *CVPR*, 2022. 3, 4, 6

[14] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010. 6

[15] Joël Fagot and Robert G. Cook. Evidence for Large Long-Term Memory Capacities in Baboons and Pigeons and its Implications for Learning and the Evolution of Cognition. *PNAS*, 103(46):17564–17567, 2006. 1

[16] Robert M. French. Catastrophic Forgetting in Connectionist Networks. *TiCS*, 3(4):128–135, 1999. 1, 2

[17] Jaime Grutzendler, Narayanan Kasthuri, and Wen-Biao Gan. Long-Term Dendritic Spine Stability in the Adult Cortex. *Nature*, 420:812–816, 2002. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 1, 6

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[20] Yu-Hsing Hsieh, Guan-Sheng Chen, Shun-Xian Cai, Ting-Yun Wei, Huei-Fang Yang, and Chu-Song Chen. Class-incremental Continual Learning for Instance Segmentation with Image-level Weak Supervision. In *ICCV*, 2023. 2

[21] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *CVPR*, 2023. 2

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. *PNAS*, 114(13):3521–3526, 2017. 2

[23] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Nav. Res. Logist.*, 2(1-2):83–97, 1955. 4

[24] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M. Ni, and Lei Zhang. DN-DETR: Accelerate DETR Training by Introducing Query DeNoising. In *CVPR*, 2022. 2

[25] Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE TPAMI*, 40(12):2935–2947, 2018. 2, 5

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 7

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *CVPR*, 2017. 4

[28] Man Liu, Chunjie Zhang, Huihui Bai, Riquan Zhang, and Yao Zhao. Cross-Part Learning for Fine-Grained Image Classification. *IEEE TIP*, 31:748–758, 2022. 1

[29] Man Liu, Feng Li, Chunjie Zhang, Yunchao Wei, Huihui Bai, and Yao Zhao. Progressive Semantic-Visual Mutual Adaption for Generalized Zero-shot Learning. In *CVPR*, 2023. 1

[30] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2018. 6

[31] Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychol. Learn. Motiv.*, pages 109–165. 1989. 2

[32] Umberto Michieli and Pietro Zanuttigh. Continual Semantic Segmentation via Repulsion-Attraction of Sparse and Disentangled Latent Representations. In *CVPR*, 2021. 2, 7

[33] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating Deep Neural Networks using Focal Loss. In *NeurIPS*, 2020. 4

[34] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *ICCV*, 2017. 1

[35] Firat Ozdemir and Orcun Goksel. Extending Pretrained Segmentation Networks with Additional Anatomical Structures. *IJCARS*, 14(7):1187–1195, 2019. 1

[36] Minh Hieu Phan, The-Anh Ta, Son Lam Phung, Long Tran-Thanh, and Abdesselam Bouzerdoum. Class Similarity Weighted Knowledge Distillation for Continual Semantic Segmentation. In *CVPR*, 2022. 2, 6, 7

[37] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive Prompts: Continual Learning for Language Models without Forgetting. In *ICLR*, 2023. 3

[38] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and and Gerald Tesauro. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *ICLR*, 2019. 2

[39] Anthony Robins. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connect. Sci.*, 7(2):123–146, 1995. 2

[40] Chao Shang, Hongliang Li, Fanman Meng, Qingbo Wu, Heqian Qiu, and Lanxiao Wang. Incrementer: Transformer for Class-Incremental Semantic Segmentation with Knowledge Distillation Focusing on Old Class. In *CVPR*, 2023. 2, 6

[41] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: COntinual Decomposed Attention-based Prompting for Rehearsal-Free Continual Learning. In *CVPR*, 2023. 2, 3

[42] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for Semantic Segmentation. In *ICCV*, 2021. 2

[43] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In *DLMIA*, 2017. 4

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017. 4

[45] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. FOSTER: Feature Boosting and Compression for Class-Incremental Learning. In *ECCV*, 2022. 3

[46] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *ECCV*, 2020. 2

[47] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers. In *CVPR*, 2021. 2

[48] Xiaoyang Wang, Bingfeng Zhang, Limin Yu, and Jimin Xiao. Hunting Sparsity: Density-Guided Contrastive Learning for Semi-Supervised Semantic Segmentation. In *CVPR*, 2023. 1

[49] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *ECCV*, 2022. 3

[50] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning To Prompt for Continual Learning. In *CVPR*, 2022. 2, 3

[51] Jiawen Xiao, Changbin Zhang, Jiekang Feng, Xialei Liu, Joost van de Weijer, and Mingming Cheng. Endpoints Weight Fusion for Class Incremental Semantic Segmentation. In *CVPR*, 2023. 2

[52] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified Perceptual Parsing for Scene Understanding. In *ECCV*, 2018. 2

[53] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *NeurIPS*, 2021. 2

[54] Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically Expandable Representation for Class Incremental Learning. In *CVPR*, 2021. 3, 6

[55] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. K-means Mask Transformer. In *ECCV*, 2022. 2

[56] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, and Yao Zhao. Credible Dual-Expert Learning for Weakly Supervised Semantic Segmentation. *IJCV*, 131:1892–1908, 2023. 2

[57] Changbin Zhang, Jiawen Xiao, Xialei Liu, Yingcong Chen, and Mingming Cheng. Representation Compensation Networks for Continual Semantic Segmentation. In *CVPR*, 2022. 2, 7

[58] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow Learner with Classifier Alignment for Continual Learning on a Pre-trained Model. In *ICCV*, 2023. 2

[59] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *ICLR*, 2023. 1, 2

[60] Zekang Zhang, Guangyu Gao, Zhiyuan Fang, Jianbo Jiao, and Yunchao Wei. Mining Unseen Classes via Regional Objectness: A Simple Baseline for Incremental Segmentation. In *NeurIPS*, 2022. 2

[61] Zekang Zhang, Guangyu Gao, Jianbo Jiao, Chi Harold Liu, and Yunchao Wei. CoinSeg: Contrast Inter- and Intra- Class Representations for Incremental Segmentation. In *ICCV*, 2023. 2

[62] Xinqiao Zhao, Feilong Tang, Xiaoyang Wang, and Jimin Xiao. SFC: Shared Feature Calibration in Weakly Supervised Semantic Segmentation. In *AAAI*, 2024. 2

[63] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing Through ADE20K Dataset. In *CVPR*, 2017. 6

[64] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning. In *ICLR*, 2023. 3

[65] Lanyun Zhu, Tianrun Chen, Jianxiong Yin, Simon See, and Jun Liu. Continual Semantic Segmentation with Automatic Memory Sample Selection. In *CVPR*, 2023. 2

[66] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*, 2021. 2