Towards Continual Universal Segmentation

Zihan Lin Zilei Wang Xu Wang
University of Science and Technology of China
myustc@mail.ustc.edu.cn, zlwang@ustc.edu.cn, xu-wang@mail.ustc.edu.cn

Abstract

*Despite the significant progress in continual image segmentation, existing arts still strive to balance between stability and plasticity. Additionally, they are specialist to specific tasks and models, which hinders the extension to more general situations. In this work, we present **CUE**, a novel **C**ontinual **U**niversal s**E**gmentation pipeline that not only inherently tackles the stability-plasticity dilemma, but unifies **any** segmentation across tasks and models as well. Our key insight: any segmentation task can be reformulated as an understanding-then-refinement paradigm, which is inspired by humans' visual perception system to first perform high-level semantic understanding, then focus on low-level vision cues. We claim three desiderata for this design: **Continuity** by inherently avoiding the stability-plasticity dilemma via exploiting the natural differences between high-level and low-level knowledge. **Generality** by unifying and simplifying the landscape towards various segmentation tasks. **Efficiency** as an interesting by-product by significantly reducing the research effort. Our resulting model, built upon this pipeline by complementary expert models, shows significant improvements over previous state-of-the-arts across various segmentation tasks and datasets. We believe that our work is a significant step towards making continual segmentation more universal and practicable.*

1. Introduction

Deep neural networks have shown great success in various computer vision tasks [23, 37, 48]. Conventionally, they are trained in a single-shot manner without considering further updates. As a result, they may lack the flexibility to handle situations that evolve over time. Typical solutions include either re-training the networks with both previous and new data or fine-tuning the networks solely on new data. Nevertheless, the former is computationally intensive, while the latter poses a great challenge of preserving previous knowledge and is known as the catastrophic forgetting [36]. With the hope of mimicking humans by gradually acquiring new concepts in a continual fashion, continual learning has been

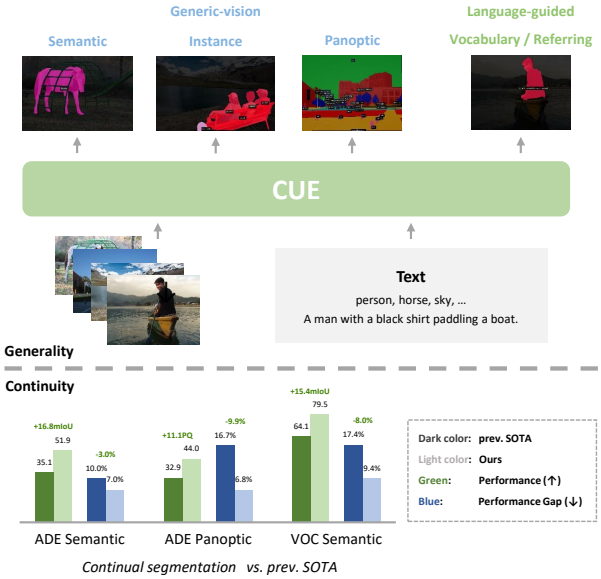


Figure 1. We present ***CUE***, a novel pipeline with key properties of generality and continuity that not only unifies segmentation across models and tasks, but sets new state-of-the-art on continual segmentation by a significant margin as well.

proposed and receives increasing attention. Recently, various approaches have been proposed for continual classification [17, 28, 39, 57, 60, 61, 72]. A few attempts extend their successes to the field of continual semantic segmentation [6, 8, 18, 53, 67, 69]. Typically, a cross-entropy loss is applied to handle new knowledge together with a regularization like knowledge distillation (KD) [27] to preserve old knowledge. How to carefully balance between stability (*i.e.*, preserve previous states) and plasticity (*i.e.*, incorporate new knowledge) becomes the main stream of current research.

Though groundbreaking, the stability-plasticity dilemma still poses a great challenge in segmentation tasks. To get a clearer understanding of why this dilemma happens, we start by examining how forgetting impacts the network in continual updates. For the first time, we reveal that the high-level knowledge (*i.e.*, semantic understanding) and the

low-level knowledge (*i.e.*, boundary prediction) show distinctive sensitivity to forgetting. Specifically, we observe that the network preserves a holistic semantic understanding across steps by correctly identifying objects in a given scene while failing to predict faithful boundaries. This inherent difference reveals a crucial reason why stability and plasticity cannot be simultaneously achieved by previous arts: A regularization that enforces enough stability for low-level knowledge could be too restrictive for high-level knowledge, excessively sacrificing its plasticity; while a proper constraint for high-level knowledge could be too flexible to effectively preserve low-level knowledge.

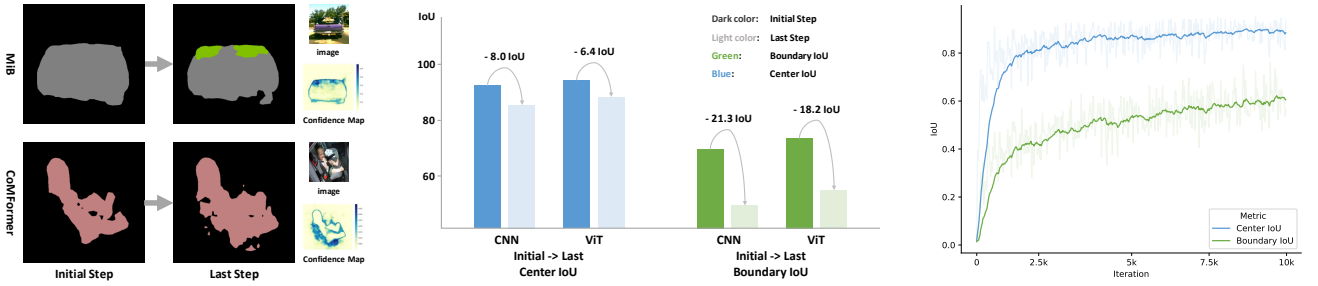
Based on this understanding and motivated by humans' visual perception system [9]. For the first time, we propose to decouple high-level and low-level knowledge by reformulating segmentation tasks into an *understanding-then-refinement* pipeline. In this pipeline, the model first focuses on high-level semantic understanding by identifying "what" and "where" of objects in a scene, then grounds them to low-level vision cues for final predictions. By this design, we can achieve better **Continuity** by simultaneously attending to both plasticity and stability instead of naively balancing between each other. This is achieved by applying suitable constraints for high-level and low-level knowledge, respectively. On top of that, it simplifies the landscape of **Generality** by unifying **any** image segmentation task into a bio-inspired pipeline. This spans across semantic, instance and panoptic segmentation, as well as complex ones like language-guided segmentation and beyond, greatly promoting the reusability and transferability of techniques under the same pipeline. Notably as a by-product, it brings better **Efficiency** by utilizing the fast convergence speed of high-level knowledge (Sec. 3.3 and Fig. 2c) to significantly reduce the computational cost. This benefit helps to pave the way towards scaling to larger networks and datasets. Our resulting model follows this novel pipeline and leverages complementary expert models to handle semantic understanding and visual refinement. Extensive experiments demonstrate that our method significantly surpasses existing works across datasets and tasks on both the aspects of higher performance and less gap to upper-bound, as shown in Fig. 1. Our contributions are summarized as follows:

- We reveal that the distinction of forgetting between high-level and low-level knowledge is a key towards simultaneous plasticity and stability, which was never explicitly considered before.
- We propose a novel segmentation pipeline by decoupling the high-level and low-level knowledge, greatly simplifying the landscape of continual universal segmentation by properties of Continuity, Generality and Efficiency.
- We conduct experiments to validate the effectiveness of the pipeline, which shows significant improvements over previous state-of-the-arts across datasets and tasks.

2. Related Work

Continual Learning. The continual learning problem has been extensively studied in image classification. The most commonly used techniques can be roughly divided into rehearsal methods and regularization methods. Rehearsal methods exploit a memory to preserve a fraction of previous training data and replay it when learning new knowledge, including replaying original images [3, 56, 57], generative replay [22, 55], intermediate feature replay [29, 72] and memory management [46, 47, 64]. Balanced finetune [5] and classifier re-balancing [28] are further introduced to ease the bias towards new categories caused by imbalance between memory data and new data. Regularization methods aim to keep the consistency between the previous network and the current one. Earlier works focus on parameter constraints [1, 36, 68] which are rarely seen recently. Modern ones apply constraints on the intermediate embeddings [15, 17, 21, 31, 65] and the output logits [28, 39, 57, 66]. These constraints are typically carefully designed KD losses to trade between stability and plasticity.
Continual Segmentation. Based on the above successes, some works extend these techniques to the field of segmentation. [6] proposes a logits distillation approach targeting background shift problem, and [53] makes improvements over it. Feature distillation [18, 42, 52, 69] is also a popular choice for better stability in continual updating. There's a bloom of replay-based methods recently, including using previous data [8], external data [51] and intermediate features [53] for replay. Among them, [73] sets a strong baseline by powering replay with reinforcement learning.
Universal Image Segmentation. Traditionally, semantic segmentation performs a per-pixel classification task [48], while instance segmentation groups pixels based on object instances [26]. Recently, panoptic segmentation [34] has been proposed to achieve a unification of them. Upon all three, the concept of universal image segmentation has emerged to unify all of them into a single framework [12, 13, 30]. Among all universal models, Maskformer [12] and Mask2former [13] are widely recognized as pioneer works, which are inspired by the success of DETR [4] handling object proposals in the scope of transformers. Following this design, more recent works [74, 75] even extend the concept of universal segmentation to vision-language tasks like open-vocabulary and referring segmentation.

Notably, recent research combining Mask2former and existing continual learning techniques has emerged to perform continual panoptic segmentation [7, 24, 33]. However, these methods are a simple combination of existing arts, which still face the stability-plasticity dilemma as previous works and is limited to generic-vision segmentation tasks. Unlike them, our work aims at inherently tacking the stability-plasticity dilemma and unifying any segmentation task under a generalized pipeline.



(a) Examples of predictions across steps.

(b) The amount of forgetting in boundary and center areas.

(c) Convergence speed of boundary and center IoU.

Figure 2. In segmentation, we define the high-level knowledge as identifying an object's category and it's rough location (*i.e.*, semantic understanding) and the low-level knowledge as predicting faithful boundaries. (a) Two examples are visualized to demonstrate that low-level knowledge suffers more on forgetting. (b, c) We measure IoU at boundary areas [11] and center areas as proxies for low-level and high-level knowledge, respectively. We validate that low-level knowledge suffers more on forgetting (b) and converges slower (c).

3. Preliminaries

3.1. Problem Definition

In continual segmentation tasks, the learning process is composed of a series of T learning steps. At each step $t \in \{1, 2, ..., T\}$, a new training set $\mathcal{D}_t = \{(x_n, y_n)\}$ is given where x_n and y_n denote an image and its annotation. A set of new categories $\mathcal{C}_t$ is introduced together with $\mathcal{D}_t$. Generally, only $\mathcal{C}_t$ is labeled in $\mathcal{D}_t$ while the others ($\mathcal{C}_{1:t-1}$ and $\mathcal{C}_{t+1:T}$) are marked as background. It's assumed that the classes at each step are disjoint, *i.e.*, $\bigcap_{t=1}^{T} \mathcal{C}_t = \varnothing$. The network is trained sequentially on each step. The goal of step t is to preserve the knowledge of $\mathcal{C}_{1:t-1}$ while acquiring the knowledge of $\mathcal{C}_t$ and finally obtain a model that performs well on all the classes seen so far.

3.2. Diving Into the Dilemma

In continual segmentation, the stability-plasticity dilemma still poses a great challenge to make it more practicable. But is there a way we can *achieve both of them simultaneously?* Bearing this question, we start by analyzing how the performance deteriorates in later updates. Without loss of generality, we opt for VOC2012 15-1 setup with 6 steps to present an analysis. The details of this setup can be found in Sec. 5.1. Two representative methods based on CNN architecture (MiB [6]) and Transformer (ViT) architecture (CoMFormer [7]) are used as examples.

We visualize two samples as in Fig. 2a to compare the models' predictions before and after continual updates. It can be observed that the models are prone to making mistakes at boundary areas after updates. Specifically, both models fail to preserve enough low-level knowledge to generate faithful boundaries when further updates are involved. On the contrary, while the boundaries are corrupted, they still recognize the objects and give correct responses at most parts of them. Especially at center areas where the models remain high confidence (measured using entropy of output logits; lighter color means lower entropy, thus higher confidence). This indicates that the models still hold a correct high-level semantic understanding of the given scene.

To further validate this conclusion beyond qualitative results, we opt for boundary IoU [11] to quantify low-level knowledge. Pixels not considered as boundaries are also measured, and we term this metric as center IoU, which measures the overall accuracy without boundary and can serve as a proxy for high-level knowledge. We compare the changes of these two metrics between the initial and last step, which is plotted in Fig. 2b. It clearly shows that after several learning steps, the boundary IoU drops significantly, indicating serious forgetting for low-level knowledge. In contrast, the center IoU decreases moderately, indicating higher robustness for high-level knowledge.

Discussion. From the pioneer experiments, it's evident that the high-level and the low-level knowledge have a natural distinction on forgetting. This is also intuitive, as it's similar to our humans' memory. Thinking about a situation that we try to recall things happened long ago. It's hard to clearly remember every detail (like predicting accurate boundaries at pixel-level), but we can still have a vague memory of them (like roughly identifying what and where of an object). This understanding helps to dive deeper into the defects of conventional segmentation pipeline. Generally, a segmentation model follows an encoder-decoder pipeline as in Fig. 3 (Upper). The main problem is that the low-level and high-level knowledge are implicitly entangled in the whole process. As a result, a constraint in favor of plasticity could be too flexible to preserve low-level knowledge, while one in favor of stability could be too rigid for handling upcoming high-level knowledge. This explains why existing arts can only balance between stability and plasticity instead of attending them simultaneously.

3.3. Towards Continual Universal Segmentation

We believe the key to boosting continual segmentation is to decouple high-level and low-level knowledge and differentiate their constraints to construct a more promising pipeline. Inspired by humans' visual perception system [9] that prioritizes global properties over details, we propose that *any image segmentation task can be reformulated as an understanding-then-refinement paradigm*. In this pipeline,

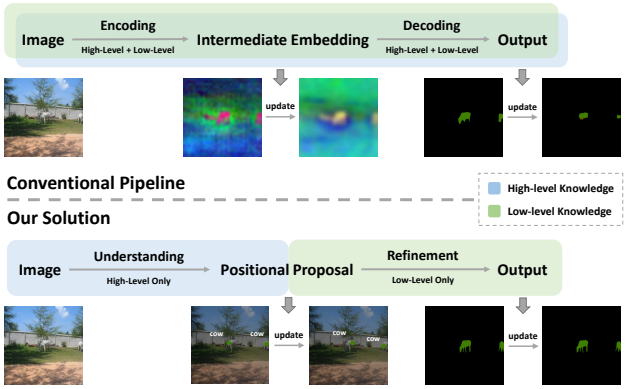


Figure 3. Illustration of the differences between conventional pipeline and our solution. We explicitly decouple high-level and low-level knowledge into semantic understanding part and visual refinement part, and use positional proposal as a more stable intermediate representation than intermediate embedding.

the model first specializes in global semantic understanding and gives class-aware positional proposals. Then it concentrates on grounding each proposal to low-level vision cues for final predictions, as shown in Fig. 3 (Lower).

To this end, we first claim its **Continuity** by effortlessly attending to both plasticity and stability instead of merely balancing between them. With this decoupled design, we can distinguish the constraints to pursue more plasticity for high-level knowledge while ensuring enough stability for low-level knowledge. Additionally, we claim its **Generality** by unifying image segmentation across tasks and models via mimicking humans' visual system. This not only provides a new insight for future research aiming at general segmentation, but also promotes the reusability and transferability of future techniques under this pipeline.

As an interesting by-product, we point out that this design helps in **Efficiency** by utilizing the fast convergence speed of high-level knowledge. We track the same metrics as in Sec. 3.2 at the training phase (specifically, the initial step of VOC 15-1 setup of CoMFormer), and plot the results in Fig. 2c. As can be seen, the center IoU quickly rises at the very early stage and significantly slows down afterward, while the boundary IoU converges much more slowly and keeps rising as more iterations are involved. Similar phenomena are also observed in generative models, which mainly focus on semantic learning at the early training stage [59]. By making the semantic understanding part handle novel categories while keeping the visual refinement part relatively fixed, we could significantly reduce the GPU hour with fewer training iterations, which holds its unique benefit when scaling up to larger networks and datasets.

Lastly, we also claim better output consistency across steps for this pipeline. Traditionally, the dense image embedding is served as an intermediate representation, from which the predictions are directly decoded. This requires pixel-level accuracy for image embedding. Any slight change could result in inconsistent outputs across steps (Fig. 3 Upper). In our pipeline, we use positional proposal as a better intermediate representation. It's more tolerant to the inevitable changes that happen in continual updating as it does not require pixel-level accuracy to represent objects, as shown in Fig. 3 (Lower). This is a desired attribute to ensure better consistency in predictions across steps.

4. Method

In this section, we introduce a framework based on our novel pipeline as in Fig. 4. Firstly, we present the semantic understanding part, which is based on a DETR-like [4] design (Sec. 4.1 to Sec. 4.3). To better fit it into continual updating, we integrate it with techniques including separated handling of things and stuff (Sec. 4.2), semantic-aware query initialization and grouped matching (Sec. 4.3). After that, we present the visual refinement part, which combines proposals with low-level visual cues to get final predictions (Sec. 4.4). Finally, we discuss how this framework is updated in a continual manner (Sec. 4.5).

4.1. Feature Extraction and Enhancement

Conventional universal works [13, 30] are limited to generic-vision tasks. In this work, we'd like to explore our pipeline beyond this to the field of language-guided segmentation. We distinguish the designs for generic-vision and language-guided tasks: In generic-vision tasks, image inputs $\mathbf{I}$ are passed to the vision encoder to generate visual embeddings $\mathbf{Z_v}$. In language-guided tasks, an additional text encoder is applied. Referring expressions or vocabularies are used as text inputs $\mathbf{T}$ to extract text embeddings $\mathbf{Z_t}$. These embeddings are then fed into the enhancer which is composed of several transformer layers with the following difference. For generic-vision tasks, we adopt standard self-attention (Self-Attn) to enhance the visual embeddings.

$$\mathbf{Z_v^{'}} = \mathbf{Z_v} + \text{Self-Attn}(\mathbf{Z_v}) \tag{1}$$

For language-guided tasks, we adopt bi-directional cross-attention (Bi-XAttn) as a common practice to make the visual and text embeddings mutually aware of each other [45].

$$(\mathbf{Z_v^{'}}, \mathbf{Z_t^{'}}) = (\mathbf{Z_v}, \mathbf{Z_t}) + \text{Bi-XAttn}(\mathbf{Z_v}, \mathbf{Z_t}) \tag{2}$$

4.2. Separating Things and Stuff

Generally in DETR-like segmentation models [12, 13, 38], things and stuff categories are processed together without distinguishing each other. While we observe that in continual updating, things and stuff categories tend to interfere with each other. We hypothesize that this might result from the difference between how network recognizes things and stuff. Things categories (*e.g.*, person, vehicles, animals) have well-defined and class-related geometry. This makes the network rely more on the shape to recognize them. For

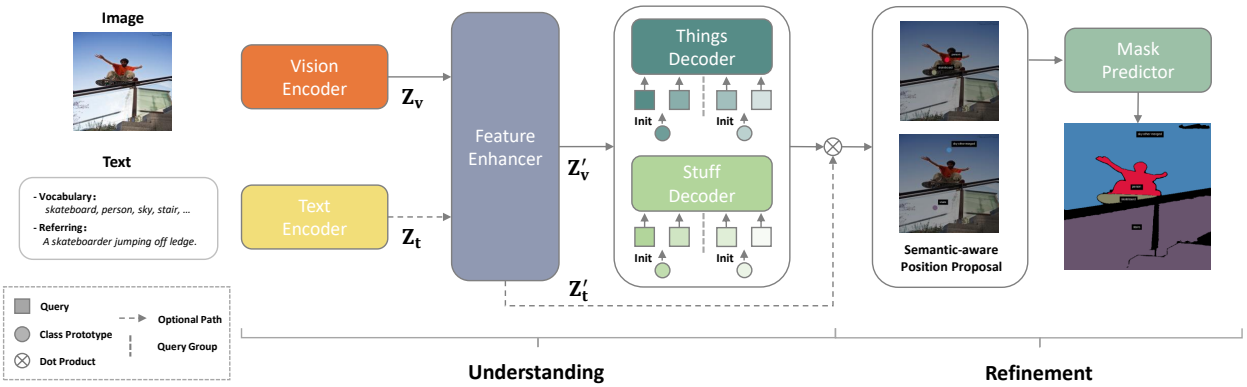


Figure 4. **Framework overview.** This framework is composed of two expert models specializing in semantic understanding and visual refinement. the understanding part follows a DETR-like [4] architecture with continual-oriented designs to generate class-aware positional proposals. The refinement part uses SAM [75] to ground each proposal to low-level details and produce final predictions.

stuff categories (*e.g.*, river, sky, mountain) that do not have a stationary geometry, the texture serves as a more important clue. This difference might forcibly distract the network from the previous recognition pattern if they are processed together in continual updates. As the decoder takes the responsibility for semantic understanding by attending queries to visual embeddings, we adopt two independent decoders for things and stuff respectively to address this.

We distinguish the two decoders as follows: The things decoder follows the design of MaskDINO [38] which consists of a mask head and a detection head. We'd like to exploit the mutual benefits of combining detection and segmentation to enhance the decoder's localization ability and ensure the outputs hit targets more accurately especially for small objects. For stuff categories, they often span across a large area which is easy to localize. We only predict masks for stuff categories using the Mask2former decoder [13].

Notably, general segmentation models need to handle low-level details to generate accurate predictions at pixel-level. This usually requires the queries to interact with multi-scale, high resolution embeddings by multiple layers of transformer decoder blocks to gradually refine the outputs, which is computationally intensive. In contrast, our goal is to only achieve a holistic semantic understanding by identifying the category and coarse position for each object in a given scene without paying attention to low-level details. Our decoders only attend to lower resolution, single-scale embeddings (*e.g.*, 1/16 image size when using ViT vision encoder with a patch size of 16) with fewer transformer decoder layers and also predict lower resolution masks. This significantly boosts the efficiency by decreasing the token length of each decoder layer and the total number of decoder layers.

The mask proposals are obtained by dot-product between object embeddings $\mathbf{Q}$ output by decoder and visual embeddings $\mathbf{Z}_\mathbf{v}^{'}$. To assign class label to mask proposals, we use cosine similarity between $\mathbf{Q}$ and text embeddings $\mathbf{Z}_\mathbf{t}^{'}$ (language-guided) or a learnable embedding $\mathbf{E}$ serving as classifier weight (generic-vision) to predict logits and assign label to proposals. For both decoders, we adopt cross-entropy loss $\mathcal{L}_{ce}$ for class prediction, binary cross-entropy loss and DICE loss [63] for mask loss $\mathcal{L}_{mask} = \lambda_{bce}\mathcal{L}_{bce} + \lambda_{dice}\mathcal{L}_{dice}$. For things decoders, L1 loss and GIoU loss [58] are additionally used as box loss $\mathcal{L}_{box} = \lambda_{L1}\mathcal{L}_{L1} + \lambda_{giou}\mathcal{L}_{giou}$ to apply to the detection head. Following DETR-like models, auxiliary losses of the same form are added after each decoder layer. Formally, the loss is computed as follows.

$$\begin{aligned} \mathcal{L}_{things} &= \lambda_{cls}\mathcal{L}_{ce} + \mathcal{L}_{mask} + \mathcal{L}_{box} \\ \mathcal{L}_{stuff} &= \lambda_{cls}\mathcal{L}_{ce} + \mathcal{L}_{mask} \\ \mathcal{L} &= \mathcal{L}_{things} + \mathcal{L}_{stuff} \end{aligned} \tag{3}$$

4.3. Query-based Micro Designs

Semantic-aware Query Initialization. Previous works [6, 18, 43, 69] have shown that a proper initialized classifier can benefit performance by decreasing the mismatch between embeddings and classifier weights. In DETR-like decoders, queries directly interact with embeddings by dot-product in cross-attention, which is very similar to how classifier weight interacts with embeddings in per-pixel classification based segmentation. We point out that proper initialized queries also provide similar benefits. Generally, there are two types of query initialization methods for DETR-like decoders, which are static query [4] and dynamic query [70]. Though the dynamic query can solve the mismatch problem by dynamically initializing queries from embeddings of current batch, the query selection is affected by the recency bias problem [28] in continual learning and causes biased initialization, which is not desired. Because of this, we adopt the static query and gradually increase the number of queries at each step with proper initialization. Due to the bipartite matching mechanism [4], each query does not strictly correspond to a category, making one-to-one initialization between new category prototypes and queries sub-optimal. We choose to initialize each new query $\mathbf{q}_{new}$ at the start of current step t based on the average of prototypes of new categories $\mathbf{p}_c$ with small Gaussian noise to differen-

tiate each other.

$$\mathbf{q}_{new} = \frac{1}{|\mathcal{C}^t|} \sum_{c \in \mathcal{C}^t} \mathbf{p}_c + \lambda_{gaussian} \mathcal{N}(0, 1) \tag{4}$$

Grouped Matching. As the queries introduced at step t are designed to represent $\mathcal{C}^t$. Instead of matching between all categories and queries as in [4, 13, 38], we group queries by learning steps and only perform matching between queries and categories belonging to the same step. This design provides two benefits: 1) It better avoids the natural instability of bipartite matching by preventing previous queries being matched to current categories and vice versa, thus avoiding unnecessary noise in optimization. 2) Unmatched queries are considered as `no-obj` and are suppressed during optimization. Due to the data imbalance problem in continual learning, previous categories are relatively rare. This could cause previous queries to be overly suppressed and result in forgetting. If all the categories within a group are not presented in input, this query group can be ignored in optimization to avoid excessive suppression on them.

4.4. Conditioned Mask Prediction

There exist several choices for handling low-level vision details. These methods range from conventional edge detectors like OWT-UCM [2], to recent segmenters like SimpleClick [44], RITM [62] and SAM [35]. In this work, we adopt the SAM to perform class-agnostic mask prediction conditioned by proposals in Sec. 4.2. The SAM decoder is designed to handle point, box, and mask input. We choose point input due to that it gives overall better results and is flexible by combining multiple points to better represent an object. The binary mask proposals obtained in Sec. 4.2 are converted to SAM point inputs as follows: We resample each proposal to a resolution of 16×16 and directly change high confidence foreground pixels to a set of points representing an object. These prompts are batched and sent to the SAM decoder to get a batch of binary masks. Each binary mask is then assigned to the category of the corresponding proposal. After that, we merge them and perform standard postprocess as in [13, 38] to obtain the final predictions.

4.5. Continual Updating

To differentiate the constraints for high-level and low-level knowledge as discussed in Sec. 3.3, we update the semantic understanding part without any KD but with only a small memory as a weak constraint. In contrast, we completely freeze the visual refinement part and utilize the transferability of low-level vision cues to handle novel categories, which is widely exploited by areas like unsupervised segmentation [32] or open-vocabulary segmentation [40].

There are two common approaches tackling the background shift [6] in continual segmentation, which are specialist KD [6, 53] and pseudo labeling [18, 42]. Given that we do not adopt KD to maximize the plasticity in semantic understanding, we use the standard pseudo labeling instead on missing old class annotations. Note that compared to existing arts that also adopt pseudo labeling, we can better avoid the risk of error accumulation caused by noisy pseudo labels via better cross-step consistency, which is discussed in Sec. 3.3 and further presented in Appendix.

5. Experiments

5.1. Implementation Details

Datasets. Following existing works, we evaluate generic-vision segmentation including semantic segmentation on PASCAL VOC 2012 [19] and ADE20K [71], and panoptic segmentation on ADE20K. We also extend to COCO [41] for additional instance and panoptic segmentation evaluation. We further explore our model on language-guided segmentation on COCO and RefCOCOg [50], which is presented in the appendix due to the space limit.

Protocols and Evaluation. There exist three different protocols for continual segmentation [52]: sequential, disjoint and overlapped. Following common practice [7, 18, 33, 53], we stick to the overlapped protocol in our benchmark as it's regarded as the most realistic and challenging one. The benchmark setup is termed as $N_o - N_n$, where N_o and N_n denote the number of classes introduced at initial step and each incremental step, respectively. For example, a 15-1 setup of a dataset containing 20 classes in total means 15 classes are given at the first step, 1 class is incrementally added per following step, resulting in 6 steps in total. The performance of old classes C_1, new classes $C_{2:T}$ and all classes $C_{1:T}$ are reported together and are denoted as *old, new, all*, respectively. The model is evaluated on the official validation sets after finishing all learning steps. The offline training results serving as a theoretical upper bound are also provided and denoted as *Joint*. Due to possibly different model architectures, we group models with different upper bound and provide their corresponding *Joint* results. To ensure a fair comparison, we additionally focus on the gap to the upper bound, which is measured as $1 - \frac{\text{perf.}}{\text{Joint}}$. We report the improvement of our model on this gap at the bottom of the table, denoted as Δ *perf. gap* (lower is better). We use the mIoU (mean Intersection-over-Union) [20], AP (average precision) [41] and PQ (panoptic quality) [34] for semantic, instance, and panoptic segmentation, respectively.

Implementation and Training. To have a comparable backbone scale to previous works using ResNet-101 [25], we adopt ViT-B [16] as the backbone with DINO-v2 [54] pre-training. For language-aware tasks, we adopt BERT-base [14] as language encoder. The feature enhancer is composed of $L = 3$ Transformer blocks. Both the things decoder and stuff decoder are composed of $L = 2$ Transformer decoder blocks. We use AdamW [49] optimizer with

Table 1. Semantic segmentation results on PASCAL VOC 2012. Our results are the average of 3 runs with standard deviation. '-M' suffix denotes method with memory. We mark the one with the least performance gap in **bold** and the second one in underline.

Method	19-1 (2 steps)			15-5 (2 steps)			15-1 (6 steps)			10-1 (11 steps)		
	old	new	all	old	new	all	old	new	all	old	new	all
MiB [6]	70.2	22.1	67.8	75.5	49.4	69.0	35.1	13.5	29.7	12.2	13.0	12.6
PLOP [18]	75.3	37.3	73.5	75.7	51.7	70.0	65.1	21.1	54.6	44.0	15.5	30.4
RCIL [69]	-	-	-	78.8	52.0	72.4	70.6	23.7	59.4	55.4	15.1	34.3
RECALL-M [51]	68.1	55.3	68.6	67.7	54.3	65.6	67.8	50.9	64.8	65.0	53.7	60.7
SSUL-M [8]	**77.8**	49.7	76.4	78.4	55.8	73.0	78.3	49.0	71.3	74.0	53.2	64.1
ALIFE-M [53]	76.7	52.2	75.5	77.6	55.2	72.3	66.0	38.8	59.5	-	-	-
Joint	77.6	77.7	77.6	79.5	71.5	77.6	79.5	71.5	77.6	78.5	76.6	77.6
AMS-M [73]	79.4	42.8	77.6	79.3	55.8	73.7	**78.5**	50.8	71.9	-	-	-
Joint	79.4	72.9	79.1	79.7	72.3	77.4	78.8	72.6	77.3	-	-	-
CUE-M (Ours)	87.6 ±0.08	**77.4** ±0.45	**87.1** ±0.12	**87.2** ±0.11	**82.8** ±0.23	**86.1** ±0.14	86.0 ±0.16	**77.9** ±0.73	**84.0** ±0.22	**83.3** ±0.19	**75.5** ±0.82	**79.5** ±0.34
Joint	87.7	89.9	87.8	87.6	88.4	87.8	87.6	88.4	87.8	87.3	88.3	87.8
Δ *perf. gap* (↓)	+0.31%	-14.95%	-1.10%	-0.05%	-15.64%	-2.90%	+1.45%	-16.93%	-2.66%	-1.16%	-15.42%	-8.04%

Table 2. Semantic segmentation results on ADE20K. Our results are the average of 3 runs with standard deviation. '-M' suffix denotes method with memory. We mark the one with the least performance gap in **bold** and the second one in underline.

Method	100-50 (2 steps)			50-50 (3 steps)			100-10 (6 steps)			100-5 (11 steps).		
	old	new	all	old	new	all	old	new	all	old	new	all
MiB [6]	40.5	17.1	32.7	45.5	21.0	29.3	38.2	11.1	29.2	36.0	5.6	25.9
PLOP [18]	41.8	14.8	32.9	48.8	20.9	30.4	40.4	13.6	31.5	39.1	7.8	28.7
RCIL [69]	42.3	18.8	34.5	48.3	25.0	32.7	39.3	17.6	32.1	38.5	11.5	29.6
SSUL-M [8]	42.7	17.5	34.3	49.1	20.1	29.7	42.8	17.6	34.4	**42.8**	17.7	34.5
ALIFE-M [53]	42.2	23.5	36.0	48.9	26.1	33.8	41.1	23.0	35.1	-	-	-
Joint	43.5	29.4	38.8	50.3	32.7	38.8	43.5	29.4	38.8	43.5	29.4	38.8
AMS-M [73]	**44.0**	24.9	37.7	-	-	-	**43.8**	25.1	37.6	43.3	18.5	35.1
Joint	44.3	28.2	39.0	-	-	-	44.3	28.2	39.0	44.3	28.2	39.0
CoMFormer [7]	44.7	26.2	38.4	-	-	-	40.6	15.6	32.3	39.5	13.6	30.9
ECLIPSE [33]	45.0	21.7	37.1	-	-	-	43.4	17.4	34.6	43.3	16.3	34.2
Joint	46.9	35.6	43.1	-	-	-	46.9	35.6	43.1	46.9	35.6	43.1
CUE-M (Ours)	**58.5** ±0.06	**46.8** ±0.24	**54.6** ±0.10	**63.6** ±0.08	**46.6** ±0.12	**52.3** ±0.10	58.0 ±0.17	**46.5** ±0.25	**54.2** ±0.20	57.9 ±0.16	**39.9** ±0.42	**51.9** ±0.22
Joint	58.9	49.7	55.8	64.7	51.4	55.8	58.9	49.7	55.8	58.9	49.7	55.8
Δ *perf. gap* (↓)	-0.00%	-5.87%	-1.19%	-0.69%	-9.42%	-6.61%	+0.40%	-4.56%	-0.72%	+0.09%	-14.68%	-3.01%

a weight decay of 5e-2. An initial learning rate of 1e-4 is used for the initial learning stage and 1e-5 for the following stages. It is decayed at 0.9 and 0.95 fractions of the training procedure by a factor of 10. We train our model for 10 epochs on VOC2012, 20 epochs on ADE20K and 15 epochs on COCO with a batch size of 16. We use a crop size of 518×518 for the understanding part, and a crop size of 1024×1024 for SAM in the visual refinement part. We set the memory size to 100 for VOC2012 and 300 for ADE20K and COCO, which is identical to [8, 73]. We use a data augmentation of random resized crop jittering between 0.5 and 2.0, random horizontal flipping and random color jittering. We use single-scale inference without data augmentation.

5.2. Main Results

Semantic Segmentation Benchmark. We evaluate our approach on continual semantic segmentation on PASCAL VOC 2012 and ADE20K due to their popularity. The results are reported in Tab. 1 and Tab. 2 respectively with four different setups. Note that among works utilizing memory (marked by '-M' suffix), RECALL and ALIFE apply a larger memory than us. AMS uses reinforcement learning to select optimal samples. These implements bring extra benefits and result in unfair comparison. From both benchmarks, we see that recent works achieve promising stability by sophisticated constraints. Some even freeze the network to pursue extreme stability (SSUL and ECLIPSE). These designs do help a lot to old class performance especially when dealing with multiple learning steps. As a trade-off, none of them achieves equal plasticity for new classes. In contrast, due to our decoupled design to apply suitable constraints to high-level and low-level knowledge separately. Our approach provides comparable stability to existing works while achieving significantly better plasticity simultaneously, which is demonstrated by the clear margin over previous SOTAs on *new* and can be consistently observed across setups and datasets.

Panoptic Segmentation Benchmark. We evaluate our approach on continual panoptic segmentation on ADE20K and present the results in Tab. 3. On this more challenging task, we observe similar conclusions as in semantic segmentation. Existing models prefer stability to effectively

Table 3. Panoptic segmentation results on ADE20K. Our results are the average of 3 runs with standard deviation. Results of [6, 18] and 50-50 results of [8] are taken from ECLIPSE. We mark the one with the least performance gap in **bold** and the second one in underline.

Method	100-50 (2 steps)			50-50 (3 steps).			100-10 (6 steps).			100-5 (11 steps).		
	old	new	all	old	new	all	old	new	all	old	new	all
MiB [6]	35.1	19.3	29.8	42.4	15.5	24.4	27.1	10.0	21.4	24.0	6.5	18.1
PLOP [18]	41.0	26.6	36.2	45.8	18.7	27.7	30.5	17.5	26.1	28.1	15.7	24.0
CoMFormer [7]	41.1	27.7	36.7	45.0	19.3	27.9	36.0	17.1	29.7	34.4	15.9	28.2
ECLIPSE [33]	41.7	23.5	35.6	46.0	20.7	29.2	41.4	18.8	33.9	41.1	16.6	32.9
Joint	43.2	32.1	39.5	50.2	34.1	39.5	43.2	32.1	39.5	43.2	32.1	39.5
CUE (Ours)	**51.4** ±0.13	**35.7** ±0.19	**46.2** ±0.14	**61.4** ±0.09	**33.8** ±0.17	**43.0** ±0.12	**50.6** ±0.21	**35.4** ±0.34	**45.5** ±0.24	**50.3** ±0.23	**31.5** ±0.55	**44.0** ±0.34
Joint	51.7	38.2	47.2	62.3	39.6	47.2	51.7	38.2	47.2	51.7	38.2	47.2
Δ *perf. gap* (↓)	-2.89%	-7.16%	-4.97%	-6.92%	-24.6%	-17.1%	-2.04%	-34.1%	-10.5%	-2.18%	-30.7%	-9.93%

preserve low-level knowledge but inevitably sacrifice plasticity. Our approach keeps the leading position across all setups and achieves substantial gain in terms of plasticity. On top of the previous results, this further demonstrates our model's ability across different tasks. Note that our approach is the only one using memory here. However, this does not impact the leading position of ours considering the memory is designed as a weaker substitute to KD (Sec. 4.5) and provides minor gain in ablation.

Table 4. Ablation study on ADE20K panoptic with 100-10 setup.

Method	100-10 (6 steps)		
	old	new	all
base model	44.8	24.5	38.0
+ memory	46.2	25.1	39.2
+ separating things and stuff	49.0	31.8	43.2
+ query initialization	49.7	33.5	44.3
+ grouped matching	**50.6**	**35.4**	**45.5**

Table 5. Computational complexity of representative methods.

Method	Arch	FLOPs	FPS	epochs
MiB [6]	CNN	134G	25.6	60
RCIL [69]	CNN	186G	9.3	60
CoMFormer [7]	ViT	586G	4.0	128
CUE (Ours)	ViT	147G	63.5	20

5.3. Ablation Study

Effect of each components. We investigate the effect of each component on ADE20K panoptic segmentation with 6 steps and present the results in Tab. 4. Starting from a base model without memory and all the techniques introduced in Sec. 4.2 and Sec. 4.3, the model already demonstrates its ability by showing comparable results to previous SOTA. This validates and highlights the superiority of our novel pipeline. By gradually adding continual oriented designs, we observe steady improvements. Among all of them, separating the decoders of things and stuff provides the most gain, validating its unique benefit in continual segmentation when handling both things and stuff. The memory only contributes a minor gain. It might because that the training data of ADE20K already contains plenty of old categories, which serve as an implicit memory.

Computational complexity. We analyze computational complexity on ADE20K semantic segmentation against two CNN-based models (MiB [6] and RCIL [69]) and a ViT-based model (CoMFormer [7]). We report floating-point operations (FLOPs), frame per second (FPS) and convergence speed (Training epochs) as in Tab. 5. Note that backward FLOPs cannot be accurately measured, we use training FPS as a proxy for forward FLOPs plus backward FLOPs. All results are obtained on the identical platform with a single NVIDIA RTX 3090 GPU using official implementations. Our model has a FLOPs on par with MiB using standard Deeplab-v3 [10], less than RCIL using a modified version and significantly less than CoMFormer based on Mask2Former. Due to that our model does not attend to high resolution embeddings, it achieves a much faster training FPS by significantly reducing the backward FLOPs, significantly reducing its training costs.

6. Conclusion and Limitation

In this work, we present CUE, which is the first step towards making continual segmentation more universal and practical. We first perform an in-depth analysis of why existing arts strive to balance between stability and plasticity. Build upon this and inspired by humans' visual perception system, we introduce a novel universal segmentation pipeline which possesses properties of continuity, generality and efficiency.

Nevertheless, we consider our work as a preliminary to continual universal segmentation. It's far from the truly universal model and needs further exploration, Still, we believe that our proposed pipeline can unleash its potential beyond what is presented in this paper, and provide valuable insights to the research community to make continual segmentation more universal and practicable.

7. Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant 62176246. This work is also supported by Anhui Province Key Research and Development Plan (202304a05020045), Anhui Province Natural Science Foundation (2208085UD17) and National Natural Science Foundation of China under Grant 62406098.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 2

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 6

[3] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, 2021. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 2, 4, 5, 6

[5] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 2

[6] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *ICCV*, 2020. 1, 2, 3, 5, 6, 7, 8

[7] Fabio Cermelli, Matthieu Cord, and Arthur Douillard. Comformer: Continual learning in semantic and panoptic segmentation. In *CVPR*, pages 3010–3020, 2023. 2, 3, 6, 7, 8

[8] Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *NIPS*, 2021. 1, 2, 7, 8

[9] Lin Chen. Topological structure in visual perception. *Science*, 218(4573):699–700, 1982. 2, 3

[10] Liang-Chieh Chen. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 8

[11] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, pages 15334–15342, 2021. 3

[12] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NIPS*, 34:17864–17875, 2021. 2, 4

[13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 2, 4, 5, 6

[14] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6

[15] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019. 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[17] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 1, 2

[18] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *CVPR*, 2021. 1, 2, 5, 6, 7, 8

[19] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011. 6

[20] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015. 6

[21] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *ECCV*, 2018. 2

[22] Rui Gao and Weiwei Liu. Ddgr: Continual learning with deep diffusion-based generative replay. In *ICML*, pages 10744–10763, 2023. 2

[23] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015. 1

[24] Yizheng Gong, Siyue Yu, Xiaoyang Wang, and Jimin Xiao. Continual segmentation with disentangled objectness learning and class recognition. In *CVPR*, pages 3848–3857, 2024. 2

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[28] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *ICCV*, 2019. 1, 2, 5

[29] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *ECCV*, 2020. 2

[30] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *CVPR*, pages 2989–2998, 2023. 2, 4

[31] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *CVPR*, 2022. 2

[32] Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X Yu. Unsupervised hierarchical semantic

segmentation with multiview cosegmentation and clustering transformers. In *CVPR*, pages 2571–2581, 2022. 6

[33] Beomyoung Kim, Joonsang Yu, and Sung Ju Hwang. Eclipse: Efficient continual learning in panoptic segmentation with visual prompt tuning. In *CVPR*, pages 3346–3356, 2024. 2, 6, 7, 8

[34] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, pages 9404–9413, 2019. 2, 6

[35] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 6

[36] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1, 2

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[38] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *CVPR*, pages 3041–3050, 2023. 4, 5, 6

[39] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2

[40] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, pages 7061–7070, 2023. 6

[41] 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*, pages 740–755. Springer, 2014. 6

[42] Zihan Lin, Zilei Wang, and Yixin Zhang. Continual semantic segmentation via structure preserving and projected feature alignment. In *ECCV*, pages 345–361. Springer, 2022. 2, 6

[43] Zihan Lin, Zilei Wang, and Yixin Zhang. Preparing the future for continual semantic segmentation. In *ICCV*, pages 11910–11920, 2023. 5

[44] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *ICCV*, pages 22290–22300, 2023. 6

[45] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4

[46] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, 2020. 2

[47] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Rmm: Reinforced memory management for class-incremental learning. *NIPS*, 2021. 2

[48] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2

[49] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[50] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 6

[51] Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *ICCV*, 2021. 2, 7

[52] Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *CVPR*, 2021. 2, 6

[53] Youngmin Oh, Donghyeon Baek, and Bumsub Ham. Alife: Adaptive logit regularizer and feature replay for incremental semantic segmentation. *NIPS*, 2022. 1, 2, 6, 7

[54] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6

[55] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *CVPR*, 2019. 2

[56] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 2

[57] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 2

[58] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019. 5

[59] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 4

[60] Yujun Shi, Kuangqi Zhou, Jian Liang, Zihang Jiang, Jiashi Feng, Philip HS Torr, Song Bai, and Vincent YF Tan. Mimicking the oracle: an initial phase decorrelation approach for class incremental learning. In *CVPR*, 2022. 1

[61] Pravendra Singh, Pratik Mazumder, Piyush Rai, and Vinay P Namboodiri. Rectification-based knowledge retention for continual learning. In *CVPR*, 2021. 1

[62] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *ICIP*, pages 3141–3145. IEEE, 2022. 6

[63] 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 *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International*

Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3, pages 240–248. Springer, 2017. 5

[64] Qing Sun, Fan Lyu, Fanhua Shang, Wei Feng, and Liang Wan. Exploring example influence in continual learning. *NIPS*, 35:27075–27086, 2022. 2

[65] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *ECCV*, pages 254–270. Springer, 2020. 2

[66] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 2

[67] Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Moin Nabi, Xavier Alameda-Pineda, and Elisa Ricci. Uncertainty-aware contrastive distillation for incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[68] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017. 2

[69] Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *CVPR*, 2022. 1, 2, 5, 7, 8

[70] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 5

[71] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 6

[72] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, 2021. 1, 2

[73] Lanyun Zhu, Tianrun Chen, Jianxiong Yin, Simon See, and Jun Liu. Continual semantic segmentation with automatic memory sample selection. In *CVPR*, pages 3082–3092, 2023. 2, 7

[74] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, pages 15116–15127, 2023. 2

[75] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *NIPS*, 36, 2024. 2, 5