A Continual Semantic Segmentation Method with Prototype Memory and Contrastive Distillation

1st Hongzhong Bai
Universiti Teknologi MARA Malaysia
Perak, Malaysia
cai908061@gmail.com

2nd Zhaoxiong Wen
Yilun Technology Co., Ltd.
Guangzhou, China
wenzhaoxiong@yisun3d.com

3rd Wenlong Xie
Yilun Technology Co., Ltd.
Guangzhou, China
WenlongXie1@outlook.com

***Abstract*—Deep learning has made impressive strides in semantic segmentation, but most existing models struggle with "catastrophic forgetting" when they are trained to recognize new classes over time. This issue becomes even more serious in privacy-sensitive areas such as medical image analysis, where retaining historical data is often not allowed. To tackle this problem, we introduce a new approach called Continual Semantic Segmentation with Prototype Memory and Contrastive Distillation (PMCD). Our method does not rely on storing example data. Instead, it builds a Prototype Memory Module that continuously updates and preserves a representative feature prototype for each class, capturing essential knowledge of previously learned categories without keeping any past images. At the same time, we develop a Contrastive Distillation Mechanism that integrates contrastive learning with knowledge distillation. This strategy encourages the updated model to maintain consistency with the previous model's predictions for old classes in the feature space, while also improving the distinction between old and newly introduced classes. Experiments conducted on the public benchmarks PASCAL VOC 2012 and ADE20K show that PMCD surpasses current state-of-the-art methods across multiple class-incremental learning scenarios, delivering a 3–5% gain in mean Intersection over Union (mIoU) and significantly lowering the forgetting rate. Overall, this work offers an effective, exemplar-free solution to catastrophic forgetting in continual semantic segmentation and paves the way for broader adoption in privacy-critical fields such as medical imaging.**

Keywords—Continual Semantic Segmentation; Prototype Memory; Contrastive Distillation; Catastrophic Forgetting; Medical Image Analysis

I. INTRODUCTION

Semantic segmentation is a core task in computer vision that focuses on assigning a semantic label to every pixel in an image, allowing for detailed, pixel-level understanding of a scene. With the rapid progress of Deep Convolutional Neural Networks (DCNNs) in recent years, semantic segmentation has advanced significantly and shown great practical value in areas such as autonomous driving, medical image analysis, and remote sensing [1]. Despite these achievements, most deep learning models are still trained under a static, "one-time" learning setup, where both the architecture and parameters remain fixed after being trained on a closed-set dataset. This "closed-world" assumption limits their usefulness in real-world environments, where new data and categories continuously appear [2].

When new classes need to be learned or when adapting to a new domain, a common strategy is to fine-tune a pre-trained model using new data. However, this often causes the model to lose previously acquired knowledge, a problem widely known as catastrophic forgetting [3]. To address this issue, Continual Learning (CL), also called incremental or lifelong learning, has gained increasing attention. The objective of CL is to enable models to continuously acquire new knowledge while retaining what they have already learned, similar to how humans learn over time [4].

Applying CL to dense prediction tasks leads to Continual Semantic Segmentation (CSS), which introduces additional challenges compared to tasks like image classification. Beyond catastrophic forgetting, CSS also faces the issue of semantic drift. In this scenario, the background class may absorb new or previously unseen objects during different learning stages, creating ambiguity and confusion in how the model interprets the background [5].

Existing CSS approaches generally fall into three categories: replay-based methods, regularization-based methods, and dynamic-architecture methods [6]. Replay-based methods reduce forgetting by storing and replaying a subset of old data while training on new tasks. Although effective, this strategy requires extra storage and is often impractical in privacy-sensitive contexts such as medical imaging [7]. Dynamic-architecture methods, on the other hand, prevent forgetting by adding new parameters for each new task. However, this leads to a steady increase in model size as tasks accumulate, making deployment difficult on devices with limited computational resources. Regularization-based methods provide an alternative that avoids storing exemplars. They use additional constraints in the loss function to restrict parameter updates and preserve previously learned knowledge. Among these, Knowledge Distillation (KD) is the most widely used technique. It transfers knowledge from an old model (teacher) to a new model (student) by encouraging the student to mimic the teacher's outputs [8].

Despite their effectiveness, KD-based methods have notable limitations. Traditional knowledge distillation mainly focuses on aligning pixel-level logits, which often overlooks the structural relationships between classes. As a result, class boundaries in the feature space may become blurred, worsening semantic drift, particularly in complex scenes where old and new classes appear together [9]. Moreover, relying solely on the "soft labels" produced by the old model makes it difficult to maintain clear separability between old and new knowledge. Over successive learning stages, this can lead to accumulated errors and significant performance degradation.

To overcome these shortcomings, we propose a novel exemplar-free framework for continual semantic

Corresponding Author: Hongzhong Bai , Universiti Teknologi MARA, 40450 Shah Alam, Selangor Darul Ehsanm, Perak, Malaysia, cai908061@gmail.com

segmentation called Prototype Memory and Contrastive Distillation (PMCD). Instead of storing high-dimensional and redundant raw images, our approach maintains a compact and representative prototype vector for each previously learned class. These prototypes capture the essential characteristics of each class and serve as distilled knowledge summaries. During training on new tasks, we introduce a contrastive distillation mechanism that combines knowledge distillation with prototype-guided contrastive learning. This mechanism not only ensures consistency between the predictions of the old and new models for previously learned classes, but also actively pulls pixel features of the same class closer to their corresponding prototype while pushing them away from prototypes of other classes. By strengthening feature separability, the proposed method effectively preserves and even enhances the structure of the feature space, enabling smoother integration of new knowledge while mitigating catastrophic forgetting.

The main contributions of this work can be summarized as follows:

- We propose PMCD, a novel exemplar-free framework for continual semantic segmentation that mitigates catastrophic forgetting through the integration of prototype memory and contrastive distillation.
- We design a Prototype Memory Module (PMM) that efficiently captures and updates the knowledge of historical classes without storing raw training data.
- We introduce a Contrastive Distillation Mechanism (CDM) that combines knowledge distillation with prototype-guided contrastive learning, significantly improving feature discriminability in scenarios where old and new classes coexist.
- Extensive experiments on multiple public benchmarks demonstrate that PMCD consistently outperforms state-of-the-art methods under various class-incremental learning settings.

The rest of this paper is structured as follows. Section 2 reviews related work on continual semantic segmentation, prototype learning, knowledge distillation, and contrastive learning. Section 3 presents the proposed PMCD framework in detail. Section 4 describes the experimental setup and reports the results and analysis. Section 5 discusses the findings, and Section 6 concludes the paper.

II. Related Work

This section reviews three research directions that are most relevant to our work: continual semantic segmentation, prototype learning for segmentation, and knowledge distillation combined with contrastive learning.

A. Continual Semantic Segmentation

Continual Semantic Segmentation (CSS) focuses on enabling a model to learn new semantic classes in a sequence while preserving performance on previously learned classes. Existing CSS methods are typically grouped into replay-based approaches and exemplar-free approaches. Exemplar-free methods can be further divided into regularization-based methods and dynamic-architecture methods [6].

Replay-based approaches are among the most effective strategies for reducing catastrophic forgetting. They work by storing a small subset of samples from earlier tasks (often called "exemplars") and replaying them alongside new task data during training [10]. A key challenge here is how to select and use exemplars efficiently. For example, iCaRL [10] introduced a nearest-mean-of-exemplars strategy for selecting representative samples. From an applied perspective, incremental learning ideas have also been extended into practical areas such as fault diagnosis under long-tailed data distributions [11]. However, replay-based methods require extra storage and are often unsuitable for privacy-sensitive settings such as medical imaging, where storing historical samples may be restricted or prohibited [7].

Dynamic-architecture methods mitigate forgetting by reducing interference between tasks, typically by adding task-specific parameters or expanding the network as new tasks arrive. For instance, Incrementer [12] adopts a transformer-based structure together with distillation for class-incremental semantic segmentation. While effective, these approaches come with a major drawback: model size and complexity grow roughly linearly with the number of tasks, which makes them difficult to deploy on resource-limited edge devices.

Regularization-based methods have become the dominant exemplar-free strategy. Rather than storing old data or changing the architecture, they add regularization terms to constrain how the model updates its parameters during new-task learning. Knowledge Distillation (KD) [8] is the central technique in this category. Learning without Forgetting (LwF) [13] was one of the earliest works to apply KD in continual learning by using the old model's outputs on new-task data as "soft labels" that guide the new model to retain knowledge of old classes. Many CSS methods build directly on this idea. MiB (Modeling the Background) [14] highlighted the semantic drift problem of the background class and used distillation to stabilize background predictions across phases. PLOP [9] further reduced forgetting by preserving the classifier weights from the previous model. SSUL [15] incorporated self-supervised learning signals to improve feature-level consistency. Related ideas have also been explored beyond closed-set segmentation, for example in transformer-based open-world instance segmentation settings [15].

Although KD-based regularization approaches can be effective, most of them rely heavily on pixel-level logits matching, which acts as a local and implicit constraint. This makes it difficult to preserve the global structure of the feature space and to maintain strong inter-class separability. As a result, performance can drop sharply when old and new classes are visually similar or when the scene contains many co-occurring categories.

B. Prototype Learning

Prototype learning originates from cognitive science and is based on representing a category using one or more "prototypes" that capture its central characteristics [16]. In deep learning, a prototype is typically defined as the mean feature vector of all samples belonging to a class, or as a learnable representative vector. Prediction can then be performed by measuring distances (or similarities) between a sample feature and the class prototypes. Compared with standard fully connected classifiers, prototype-based formulations provide a more interpretable, metric-driven decision mechanism and often exhibit improved generalization.

Prototype learning has been applied broadly across vision problems. In few-shot learning, Prototypical Networks [17] classify query samples by comparing them to prototypes computed from a support set. In semantic segmentation, recent work has explored replacing conventional pixel-wise classifiers with prototype-based decision rules. For example, Zhou et al. [1] re-examined semantic segmentation through a prototype perspective and later formalized this direction into a prototype-based segmentation framework [18].

In continual learning, prototypes are also attractive as a compact way to represent old knowledge. PASS [19] proposed maintaining prototypes for old classes and combining prototype augmentation with self-supervision to reduce forgetting. However, most prototype-based continual learning research has focused on image classification. Extending these ideas to continual semantic segmentation—where predictions must be made densely at the pixel level, and where class coexistence and background drift are major concerns—remains challenging. In particular, how to integrate prototype representations with mainstream CSS strategies such as knowledge distillation is still not fully resolved.

C. Contrastive Distillation

Knowledge distillation aims to transfer knowledge from a "teacher" model to a "student" model. Classic KD [8] mainly performs this transfer by aligning the teacher's and student's logits. While effective, logits matching only communicates the final prediction behavior and does not directly leverage the rich intermediate feature information inside the teacher.

To address this limitation, researchers introduced ideas from contrastive learning into distillation, forming a line of work often referred to as contrastive distillation. The central idea is that the student should not only match the teacher's outputs, but should also learn the structure of the teacher's representation space—specifically, the similarity relationships among samples. Contrastive Representation Distillation (CRD) [20] is a representative example. Earlier contrastive representation learning methods, such as Contrastive Predictive Coding (CPC) [21], provided important foundations for these distillation strategies.

In semantic segmentation, encoder–decoder architectures with atrous convolutions remain strong and widely used backbones in both static segmentation and continual segmentation systems [22]. At the same time, improvements in general deep learning architectures—supported by foundational work such as deep residual learning [23]—have played an important role in strengthening modern segmentation and continual learning models.

Within continual learning, contrastive distillation is particularly promising because it can explicitly constrain the student's representation space to stay aligned with the teacher's feature space, which helps preserve old knowledge more robustly than logits-only supervision. Still, effectively combining contrastive distillation with prototype learning—while scaling to the dense, complex nature of continual semantic segmentation—remains an open problem. The PMCD framework proposed in this paper explores this direction by introducing prototype-guided contrastive distillation, aiming to deliver a stronger and more reliable exemplar-free solution for continual semantic segmentation.

III. METHOD

To effectively address catastrophic forgetting in continual semantic segmentation without retaining historical data, we introduce a new framework called Prototype Memory and Contrastive Distillation (PMCD). In this section, we describe the overall architecture, its main components, and the training procedure in detail.

A. Overall Framework

PMCD builds upon a conventional knowledge distillation framework and extends it with two key components: a Prototype Memory Module (PMM) and a Contrastive Distillation Mechanism (CDM).

Assume that after completing the first $t-1$ learning phases, the model has learned to segment a set of classes C_{old} with parameters θ_{t-1}. In the current phase ttt, the model receives a new dataset D_t containing previously unseen classes C_{new}. The objective is to train an updated model θ_t that can accurately segment the new classes C_{new} while retaining strong performance on the previously learned classes C_{old}. Importantly, this must be achieved without accessing any data from earlier datasets $D_1, \ldots, D_{t-1}$.

The overall structure of PMCD (illustrated in Figure 1) follows a teacher–student paradigm. During phase t, the new model θ_t (student) takes images from the current task as input. The previous model θ_{t-1} serves as the teacher and remains fixed. Training is guided by three complementary loss functions:

1) Standard Segmentation Loss (L_{seg}): For pixels belonging to the new classes C_{new}, we apply the standard cross-entropy loss to provide direct supervised learning signals.

2) Knowledge Distillation Loss (L_{kd}): For pixels corresponding to the old classes C_{old} and the background, the soft predictions produced by the teacher model θ_{t-1} are used to supervise the student model. This encourages the new model to preserve previously acquired knowledge.

3) Prototype Contrastive Loss (L_{proto}): This loss leverages the stored prototypes of old classes maintained in the Prototype Memory Module (PMM). Through contrastive learning, it explicitly enforces feature-level consistency for old classes while improving the separability between old and new classes in the embedding space.

The overall training objective is defined as a weighted combination of these three losses:

$$L_{total} = L_{seg} + \lambda_{kd} L_{kd} + \lambda_{proto} L_{proto} \quad (1)$$

where λ_{kd} and λ_{proto} are balancing coefficients that control the relative contributions of the distillation and prototype contrastive terms.

By integrating segmentation supervision, teacher–student consistency, and prototype-guided contrastive constraints into a unified objective, PMCD enables stable knowledge retention while facilitating the effective incorporation of newly introduced classes.

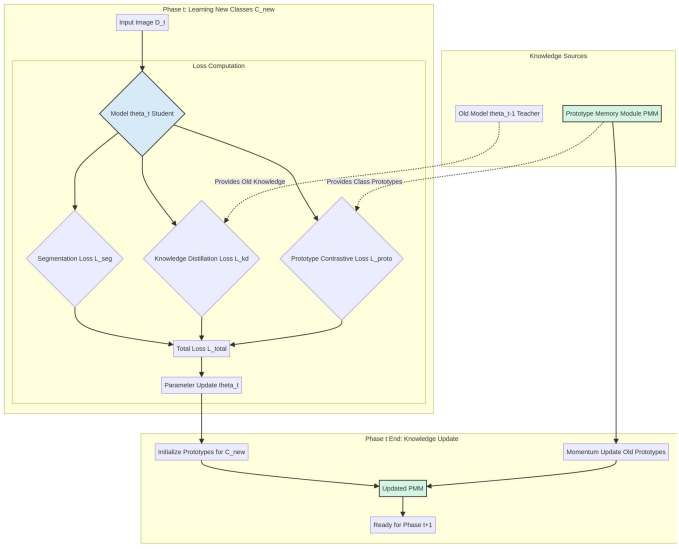

Fig. 1. Overall Framework of the PMCD Method.

B. Prototype Memory Module (PMM)

The Prototype Memory Module (PMM) is a central component of PMCD. Its purpose is to maintain a compact yet highly representative prototype vector for each semantic class the model has learned so far. In this way, the model preserves essential knowledge about historical classes without retaining any raw training images, which is particularly important in privacy-sensitive scenarios.

1) Prototype Initialization and Storage

At the end of each learning phase t_1 , once the model θ_t, has been trained on the current dataset D_t , we construct prototypes for the newly introduced classes C_{new}.

Specifically, for each class $c \in C_{new}$, we pass all images in D_t through the trained model θ_t and extract the deep feature vectors corresponding to pixels labeled as class c. Let $\mathcal{F}_c = \{f_i^c\}_{i=1}^{N_c}$ denote the set of feature vectors for all N_c pixels belonging to class c. The prototype p_c is computed as the mean of these feature vectors:

$$p_c = \frac{1}{N_c} \sum_{i=1}^{N_c} f_i^c \quad (2)$$

This averaging operation produces a single, low-dimensional vector that captures the central tendency of the feature distribution for class c. The resulting prototype serves as a condensed representation of that class's semantic characteristics in the feature space.

After initialization, each prototype p_c is stored in the Prototype Memory Module. As learning proceeds to future phases, these stored prototypes are used to guide contrastive distillation, helping the model preserve inter-class structure and reduce forgetting without requiring access to any previously seen images.

Here, S_c denotes the set of all pixels labeled as class ccc in the dataset, and $f_{\theta_t}(x_i)$ represents the feature vector of pixel iii extracted by the backbone network of model θ_t. The resulting prototype vectors are then stored in a dedicated Prototype Memory Bank, which serves as a compact knowledge repository for use in future incremental learning stages.

2) Prototype Update

When the training process moves to the next phase $t+1$, the model parameters are updated from θ_t to θ_{t+1}. As a result, the feature space undergoes a shift. If the stored prototypes remain unchanged, they may gradually become misaligned with the updated feature representations, reducing their effectiveness.

To maintain consistency between the prototypes and the evolving feature space, we adopt a momentum-based update strategy. For each old class $c \in C_{old}$, we proceed as follows: when a pixel in the new dataset is predicted by the old model θ_t to belong to class ccc, we treat it as a reliable pseudo-label. We then extract its feature representation using the new model θ_{t+1}. Let this newly extracted feature be denoted as $\tilde{f}_c$. The prototype p_c is updated using an exponential moving average:

$$p_c \leftarrow \alpha p_c + (1-\alpha)\tilde{f}^c \quad (3)$$

where $\alpha \in [0,1]$ is a momentum coefficient that controls the smoothness of the update. This gradual update mechanism allows the prototypes to track the dynamics of the feature space, ensuring they remain representative over time.

C. Contrastive Distillation Mechanism (CDM)

The Contrastive Distillation Mechanism (CDM) is the second core component of PMCD. It extends conventional knowledge distillation by integrating prototype-guided contrastive learning, enabling the model to address catastrophic forgetting and semantic drift more effectively at the feature representation level.

1) Knowledge Distillation Loss

For each pixel i in an input image whose ground-truth label does not belong to the newly introduced classes C_{new}—that is, it belongs to an old class in C_{old} or to the background—we apply a knowledge distillation constraint.

Specifically, the previous model θ_{t-1} serves as the teacher and generates soft probability outputs over the old classes. These soft predictions encode richer inter-class information than hard labels. The updated model θ_t (student) is trained to align its predictions with those of the teacher by minimizing the Kullback–Leibler (KL) divergence between their output distributions.

Let z_i^{t-1} and z_i^t denote the logits of the teacher and student models for pixel iii, respectively. After applying a temperature-scaled softmax function with temperature T, the distillation loss for old classes can be formulated as:

$$L_{kd} = \frac{1}{|\Omega_{old}|} \sum_{i \in \Omega_{old}} K\,L(\sigma(z_i^{t-1}/T) \| \sigma(z_i^t/T)) \quad (4)$$

where:

- Ω_{old} denotes the set of pixels belonging to old classes or background,

- $\sigma(\cdot)$ is the softmax function,
- T is the temperature parameter used to soften the probability distribution, and
- $KL(\cdot \| \cdot)$represents the KL divergence.

By minimizing this loss, the student model is encouraged to preserve the teacher's prediction behavior for previously learned categories. However, unlike traditional distillation approaches that rely solely on logits alignment, PMCD further strengthens knowledge retention by introducing prototype-guided contrastive learning at the feature level, which we describe next.

Here, z_i^{t-1} and z_i^t denote the logits produced by the old (teacher) and new (student) models for pixel i, respectively. The function $\sigma(\cdot)$ represents the Softmax operation, and T is the temperature parameter used in distillation to soften the probability distribution and reveal richer relational information among classes.

2) Prototype Contrastive Loss

The Prototype Contrastive Loss forms the core of our proposed method. Its goal is to explicitly impose the principle of intra-class compactness and inter-class separability within the feature space by leveraging the stored prototypes in the Prototype Memory Module (PMM).

For a pixel iii that is predicted as belonging to an old class $c \in C_{old}$, we extract its feature representation $f_{\theta_t}(x_i)^\dagger$ using the new model. This feature serves as the anchor in the contrastive learning framework. The corresponding class prototype pcp_cpc is treated as the positive sample, while all other stored prototypes $\{p_j | j \in C_{old}, j \neq c\}$ act as negative samples (as illustrated in Figure 2).

To implement this objective, we adopt the InfoNCE loss [21], which is widely used in contrastive learning. The prototype contrastive loss for a pixel iii can be written as:

$$L_{proto}^{(i)} = -\log \frac{\exp(\mathrm{sim}(f_{\theta_t}(x_i), p_c)/\tau)}{\sum_{j \in C_{old}} \exp(\mathrm{sim}(f_{\theta_t}(x_i), p_j)/\tau)} \quad (5)$$

where:

- $\mathrm{sim}(\cdot, \cdot)$ denotes a similarity function (e.g., cosine similarity),
- Tis a temperature parameter that controls the concentration level of the distribution, and
- the denominator sums over all old-class prototypes stored in the PMM.

The overall prototype contrastive loss is obtained by averaging over all eligible pixels:

$$L_{proto} = \frac{1}{|\Omega_{old}|} \sum_{i \in \Omega_{old}} L_{proto}^{(i)} \quad (6)$$

This formulation pulls features of old-class pixels closer to their corresponding prototypes while pushing them away from other class prototypes. As a result, the feature space remains well-structured even as new classes are introduced, significantly reducing semantic drift and strengthening resistance to catastrophic forgetting.

Here, $\mathrm{sim}(\cdot, \cdot)$ represents cosine similarity, which measures the angular closeness between two feature vectors, and τ is a temperature coefficient that controls how sharply similarities are distributed after scaling. A smaller τ produces a more concentrated distribution, strengthening the contrast between positive and negative pairs.

This loss encourages the pixel feature $f_{\theta_t}(x_i)$ to move closer to its corresponding class prototype p_c (intra-class compactness), while simultaneously pushing it away from other class prototypes (inter-class separability). As a result, the embedding space maintains a clear and well-organized class structure.

Importantly, this mechanism allows the model to preserve the geometric structure of previously learned classes without requiring access to the original training images. By anchoring pixel features to stable class prototypes and enforcing discriminative separation, the method effectively reduces semantic drift and minimizes confusion between old and newly introduced categories. This strengthens knowledge retention at the representation level, complementing the pixel-level supervision provided by knowledge distillation.

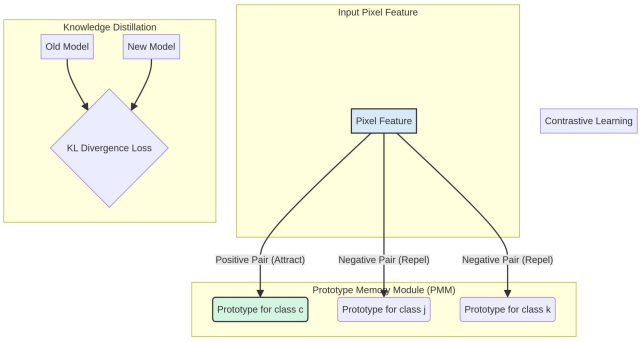

Fig. 2. Diagram of the Contrastive Distillation Mechanism.

D. Algorithmic Procedure

The full training workflow of PMCD is summarized in Algorithm 1. In each incremental phase, the model is trained on the current dataset using a combined objective that includes the standard segmentation loss, the knowledge distillation loss, and the prototype contrastive loss. Once training is complete, the model computes prototypes for the newly introduced classes and updates the prototypes of previously learned classes using the momentum strategy. This ensures that the prototype memory remains aligned with the evolving feature space and is ready for the next learning phase.

1) Algorithm 1: PMCD Training Procedure

a) Input:

- Current task dataset D_t,
- Old model parameters θ_{t-1},
- Prototype memory bank $P = \{p_c | c \in C_{old}\}$

b) Output:

- Updated model parameters θ_t,
- Updated prototype memory bank P_t

c) Initialize the new model: $\theta_t \leftarrow \theta_{t-1}$

d) For each training epoch:

For each mini-batch (X,Y) sampled from D_t:

- Compute features and predictions of the new model: $F_t, Z_t = \text{Model}(X; \theta_t)$
- Compute predictions of the old model: $Z_{t-1} = \text{Model}(X; \theta_{t-1})$
- Compute segmentation loss for pixels where $Y \in C_{new}$: $L_{seg} = \text{CrossEntropy}(Z_t, Y)$
- Compute distillation loss for pixels where $Y \notin C_{new}$: $L_{kd} = \text{KL}(Z_{t-1}, Z_t)$
- Compute prototype contrastive loss for pixels where $Y \in C_{old}$: $L_{proto} = \text{InfoNCE}(F_t, P)$
- Compute total loss: $L_{total} = L_{seg} + \lambda_{kd} L_{kd} + \lambda_{proto} L_{proto}$
- Backpropagate and update θ_t

e) Prototype memory update:

- Initialize $P_t \leftarrow P$
- For each class $c \in C_{new}$:
- Compute and store new prototype: $p_c = \text{Mean}(f_{\theta_t}(x_i))$ $\text{forall where} y_i = c$
- Update memory: $P_t \leftarrow P_t \cup \{p_c\}$
- For each class $c \in C_{old}$:
- Update prototype with momentum: $p_c \leftarrow \alpha p_c + (1 - \alpha)\text{Mean}(f_{\theta_t}(x_i))$ using relevant pixels identified in the current phase

f) Return θ_t and P_t

Through the coordinated use of prototype memory and contrastive distillation, PMCD establishes a continual learning framework that balances stability (retaining prior knowledge) and plasticity (acquiring new knowledge), all without storing historical images. Knowledge distillation preserves the predictive behavior of the previous model, while prototype-guided contrastive learning reinforces the structural integrity of the feature space. Together, these mechanisms provide a more principled and effective solution to continual semantic segmentation, offering a fresh perspective on mitigating catastrophic forgetting in dense prediction tasks.

IV. Experiments

To thoroughly assess the effectiveness of the proposed PMCD framework, we conducted extensive quantitative and qualitative experiments. This section introduces the datasets and experimental settings, implementation details, evaluation metrics, and comparisons with state-of-the-art approaches, followed by ablation studies.

A. Datasets and Experimental Settings

1) Datasets

We evaluate our method on two widely used public benchmarks to simulate different continual learning scenarios:

- PASCAL VOC 2012: A standard semantic segmentation benchmark containing 20 foreground classes and 1 background class. Following common practice, we use the augmented training set (10,582 images) for training and the validation set (1,449 images) for evaluation.
- ADE20K: A large-scale scene parsing dataset with 150 semantic categories. We use the official training set (20,210 images) and validation set (2,000 images). The large number of categories and complex scene compositions make ADE20K particularly challenging for continual learning methods.

2) Incremental Learning Protocol

We adopt the class-incremental learning setting, one of the most challenging continual learning scenarios. In this setting, new classes are introduced in successive phases. During testing, the model must recognize all classes learned so far, including both old and newly added ones.

To simulate different learning schedules, we design multiple incremental splits:

- VOC 15-1: The model is first trained on 15 classes, followed by 5 incremental phases, each introducing 1 new class.
- VOC 10-5-5: The model first learns 10 classes, then 5 additional classes in the second phase, and finally the remaining 5 classes in the last phase.
- ADE20K 100-50: The model is initially trained on 100 classes and then incrementally learns the remaining 50 classes in a subsequent phase.

These configurations allow us to evaluate performance under both gradual and large-step incremental scenarios.

B. Implementation Details

1) Network Architecture

We adopt DeepLabv3+ [22] as the segmentation framework. For reproducibility on commonly available hardware, we use a resource-friendly configuration with a ResNet-50 backbone pre-trained on ImageNet. To further evaluate the upper-bound performance, we also conduct experiments using a heavier ResNet-101 backbone [23].

The backbone outputs a high-dimensional feature representation (2048 dimensions for ResNet-101, and correspondingly lower for ResNet-50). The stored prototype vectors share the same dimensionality as the selected backbone's feature output.

2) Training Setup

All experiments are reproducible on a single consumer-grade GPU (e.g., 24GB memory). To ensure efficiency, we employ mixed-precision training and, when necessary, gradient accumulation to maintain an effective batch size under limited memory.

- Optimizer: AdamW
- Initial learning rate: 1×10^{-4}
- Learning rate schedule: Polynomial ("poly") decay
- Training epochs per incremental phase: 50

a) Batch size:

- 16 for PASCAL VOC

- 8 for ADE20K (Smaller mini-batches with gradient accumulation are used when memory is constrained.)

b) Hyperparameters

Based on a lightweight pilot search on a fixed validation split:

- $\lambda_{kd} = 1.0$
- $\lambda_{proto} = 0.5$
- Distillation temperature T=2.0
- Contrastive temperature $\tau = 0.1$
- Prototype momentum coefficient $\alpha = 0.999$

These values provide a stable balance between segmentation learning, knowledge preservation, and feature-space discrimination.

C. Evaluation Metrics

We evaluate performance using two widely adopted metrics:

- Mean Intersection over Union (mIoU): The standard metric for semantic segmentation. It computes the average IoU across all classes. We report the overall mIoU over all classes learned up to the current phase.
- Forgetting Rate (FR) [14]: This metric quantifies how much performance on previously learned classes degrades over subsequent phases. It is calculated as the average performance drop of each class after its initial learning phase. A lower FR indicates stronger resistance to catastrophic forgetting.

D. Comparative Experiments

We compare PMCD with several representative continual semantic segmentation methods:

- Fine-tuning: Directly fine-tuning the model on new data without any forgetting mitigation mechanism. This serves as a lower performance bound.
- LwF [13]: A classic knowledge distillation-based continual learning approach.
- MiB [14]: A KD-based method tailored for continual semantic segmentation, emphasizing background modeling.
- PLOP [9]: A distillation-based method that preserves old knowledge by freezing classifier weights and applying additional constraints.
- SSUL [15]: A method combining knowledge distillation with self-supervised learning to enhance feature consistency.

Through these comparisons, we demonstrate that PMCD achieves superior performance in terms of both mIoU and forgetting rate across multiple incremental settings, validating the effectiveness of prototype-guided contrastive distillation for exemplar-free continual semantic segmentation.

TABLE I. QUANTITATIVE RESULTS ON THE PASCAL VOC 2012 DATASET (UNLESS OTHERWISE STATED, THE MAIN RESULTS ARE OBTAINED WITH THE RESNET-101 BACKBONE [23], AND A RESOURCE-FRIENDLY RESNET-50 CONFIGURATION IS ALSO PROVIDED FOR REPRODUCTION AND VERIFYING CONSISTENT TRENDS).

Method	Setting	mIoU (%)	Forgetting Rate (%)
Fine-tuning	15-1	45.2	25.8
LwF	15-1	58.9	15.3
MiB	15-1	62.5	12.1
PLOP	15-1	64.8	10.5
SSUL	15-1	65.1	10.2
PMCD (Ours)	15-1	68.8	7.9
Fine-tuning	10-5-5	50.1	21.4
LwF	10-5-5	63.4	12.9
MiB	10-5-5	66.2	10.8
PLOP	10-5-5	68.1	9.1
SSUL	10-5-5	68.5	8.8
PMCD (Ours)	10-5-5	71.9	6.5

TABLE II. QUANTITATIVE RESULTS ON THE ADE20K DATASET (UNLESS OTHERWISE STATED, THE MAIN RESULTS ARE OBTAINED WITH THE RESNET-101 BACKBONE [23], AND A RESOURCE-FRIENDLY RESNET-50 CONFIGURATION IS ALSO PROVIDED FOR REPRODUCTION AND VERIFYING CONSISTENT TRENDS).

Method	Fine-tuning	LwF	MiB	PLOP	SSUL	PMCD (Ours)
Setting	100-50	100-50	100-50	100-50	100-50	100-50
mIoU (%)	28.4	35.1	37.2	38.9	39.5	42.8
Forgetting Rate (%)	30.1	20.5	18.2	16.9	16.1	13.5

The quantitative results in Table I and Table II clearly show that our proposed PMCD method surpasses all competing methods in final mIoU across various incremental

settings on both PASCAL VOC and ADE20K. For instance, in the challenging VOC 15-1 setting, PMCD's final mIoU is 3.7 percentage points higher than the next-best method, SSUL. Concurrently, PMCD's forgetting rate is also the lowest, demonstrating its superior ability to combat catastrophic forgetting. The advantage of PMCD is even more pronounced on the more complex ADE20K dataset, indicating its better scalability in large-scale continual learning scenarios (Figure 3-4).

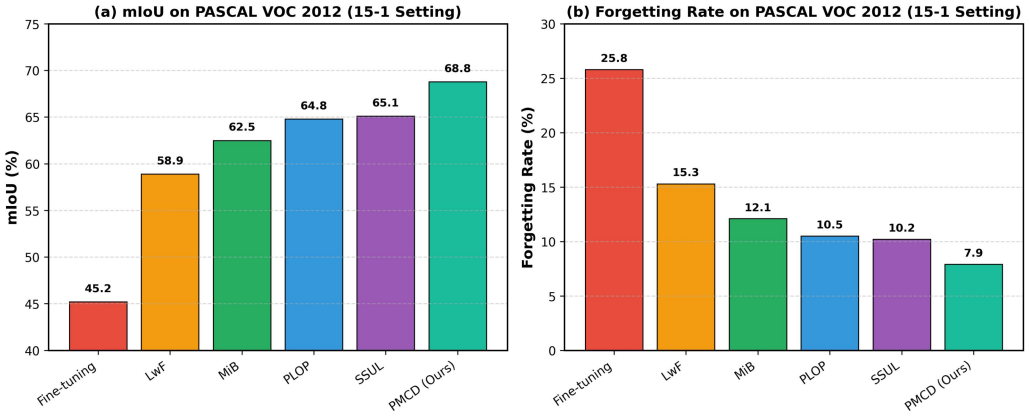

Fig. 3. Comparison of mIoU and Forgetting Rate on PASCAL VOC 2012 (15-1 setting).

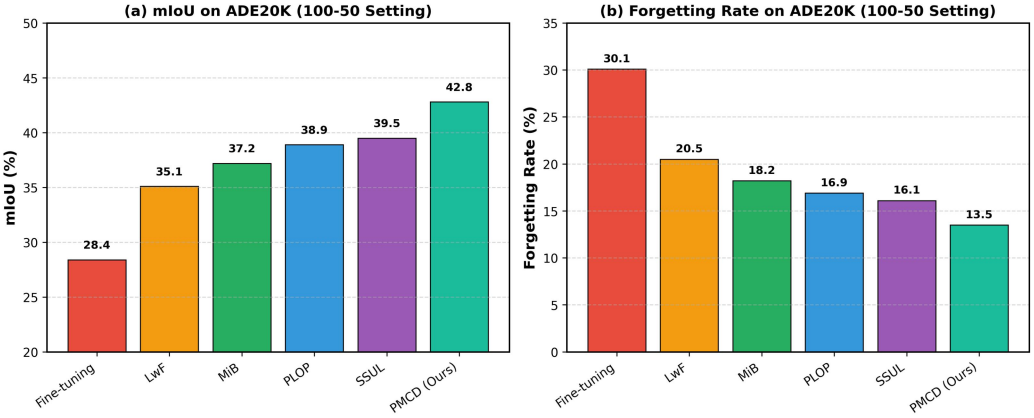

Fig. 4. Comparison of mIoU and Forgetting Rate on ADE20K (100-50 setting).

E. Ablation Study

To better understand the contribution of each component in PMCD, we performed a set of ablation studies under the VOC 15-1 incremental setting. The baseline model follows a standard knowledge distillation framework, optimized using only the segmentation loss L_{seg} and the distillation loss L_{kd}. On top of this baseline, we progressively introduced different components of PMCD to analyze their individual and combined effects.

The evaluated variants are described as follows:

- Baseline (KD only): A conventional distillation-based continual segmentation model using $L_{seg} + L_{kd}$. This serves as the reference point for all comparisons.
- + PMM (w/o update): This variant augments the baseline with the Prototype Memory Module (PMM). Class prototypes are computed after their initial learning phase and stored in memory. However, in subsequent incremental phases, these prototypes remain fixed and are not updated. The stored static prototypes are used solely for computing the prototype contrastive loss. This setting evaluates the impact of introducing prototype-based structural constraints without adapting them to feature space shifts.
- + PMM (w/ update): Building upon the previous configuration, this version incorporates the momentum-based prototype update mechanism. During later incremental phases, stored prototypes are gradually updated using features extracted by the new model. This experiment examines whether dynamically aligning prototypes with the evolving feature space improves knowledge retention and representation consistency.
- PMCD (Full Model): The complete framework, integrating both the Prototype Memory Module (PMM) and the Contrastive Distillation Mechanism (CDM). This version includes prototype initialization, momentum-based updates, and prototype-guided contrastive learning alongside knowledge distillation. It represents the final and fully functional PMCD system.

Through these ablation experiments, we systematically evaluate how static prototypes, adaptive prototype updates, and full contrastive distillation each contribute to mitigating catastrophic forgetting and enhancing feature discriminability in continual semantic segmentation.

TABLE III. ABLATION STUDY RESULTS ON THE VOC 15-1 SETTING

Method	mIoU (%)	Forgetting Rate (%)
Baseline (KD)	66.5	9.8
+ PMM (w/o update)	67.3	9.1
+ PMM (w/ update)	68.2	8.5
PMCD (Full)	68.8	7.9

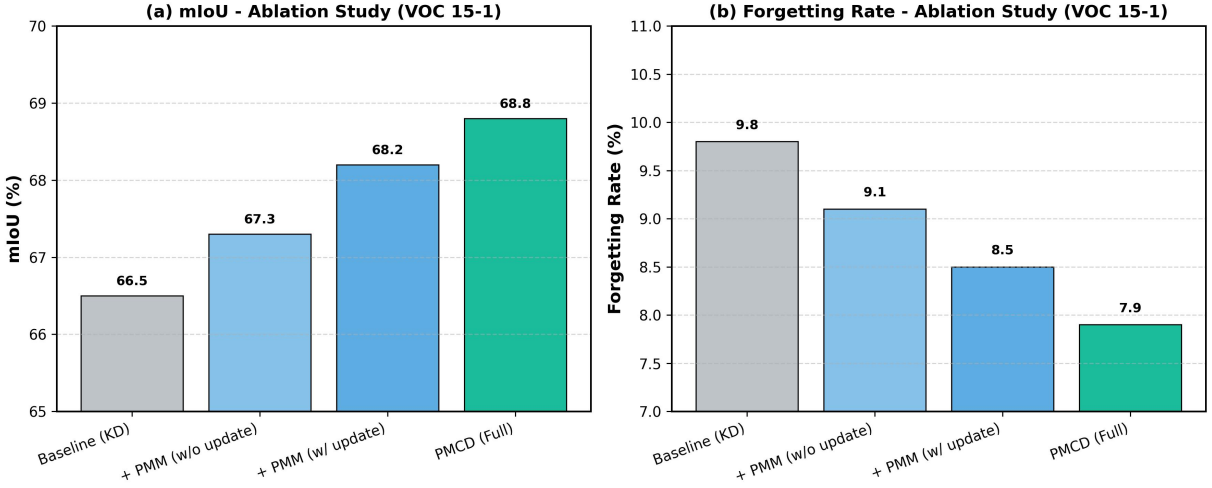

Fig. 5. Ablation Study Results.

The ablation study results (Table III) clearly reveal the contribution of each component. Simply adding static prototypes (+PMM w/o update) brings a performance gain (mIoU from 66.5% to 67.3%), indicating that compressing class knowledge into prototypes is an effective memory retention strategy in itself. With the addition of the prototype update mechanism (+PMM w/ update), performance improves further (mIoU reaches 68.2%), demonstrating the importance of adapting to feature space drift. Finally, the full PMCD model achieves the best performance (mIoU of 68.8% and forgetting rate of 7.9%), confirming that the synergy between the prototype memory and the contrastive distillation mechanism is key to its superior performance (Figure 5).

V. DISCUSSION

The results presented earlier clearly show that PMCD achieves state-of-the-art performance across multiple continual semantic segmentation benchmarks. In this section, we further analyze the reasons behind these improvements, reflect on the broader research value of the method, discuss its limitations, and outline several promising directions for future work (see Figure 6).

A. Interpretation, Research Value, and Limitations

1) Interpretation of Results

The central strength of PMCD lies in the synergy between prototype memory and contrastive distillation, which together address catastrophic forgetting and semantic drift at a structural level.

Traditional distillation-based approaches, such as LwF and MiB, primarily focus on aligning the pixel-level logits between teacher and student models. While this helps retain prediction behavior for old classes, it remains a relatively passive and local constraint. It enforces consistency in outputs but does not explicitly regulate the geometry of the feature space. As a result, the internal representation of old classes can gradually drift as new classes are introduced.

PMCD, by contrast, adopts a more active and global strategy. The Prototype Memory Module compresses each old class into a compact and representative prototype vector. This serves as an efficient abstraction of historical knowledge. The ablation study (Table III) shows that even static prototypes already yield measurable performance gains, confirming that prototype-based supervision provides meaningful structural guidance.

More importantly, the Contrastive Distillation Mechanism treats prototypes as anchors in the embedding space. By pulling pixel features toward their corresponding class prototype and pushing them away from other prototypes, PMCD explicitly enforces intra-class compactness and inter-class separability. This reshapes the feature distribution in a principled manner:

- Old class features remain tightly clustered and discriminative.
- New class features are prevented from overlapping with old class regions.
- Class confusion and semantic drift are significantly reduced.
- Rather than merely preserving output similarity, PMCD stabilizes the structure of the feature space itself.

The momentum-based prototype update mechanism further strengthens this effect. Since the feature space evolves during incremental training, fixed prototypes would eventually become misaligned with the updated representations. The ablation results confirm that enabling momentum updates significantly improves performance. By allowing prototypes to evolve smoothly with the model, they function as dynamic landmarks that guide representation learning without becoming obsolete.

B. Research Value

1) Theoretical Contribution

From a theoretical standpoint, this work proposes a new paradigm for exemplar-free continual learning. It shifts the focus from implicit, pixel-level logit alignment to explicit, metric-based preservation of feature space structure. By maintaining a compact prototype memory bank and integrating it with contrastive learning, PMCD achieves performance comparable to some replay-based methods—without storing any historical samples. This demonstrates that structured representation preservation can serve as a powerful alternative to data replay.

2) Practical Significance

Practically, PMCD is particularly valuable in privacy-sensitive domains. Because it does not require storing past data, it is well suited for applications such as:

- Medical imaging, where models may need to learn new disease categories without retaining historical patient data.
- Autonomous driving, where new object categories appear over time and data privacy regulations may restrict storage.
- Personal devices and edge systems, where user data must remain protected.

In such scenarios, PMCD supports the development of intelligent systems capable of "lifelong learning" while respecting strict data privacy constraints.

C. Limitations

Despite its strengths, PMCD has several limitations.

1) Dependence on Prototype Quality: The effectiveness of the method relies on the representativeness of the prototype memory bank. Although the momentum update strategy mitigates misalignment, prototypes may still degrade under severe domain shifts or drastic task changes.

2) Computational Overhead: The prototype contrastive loss requires computing similarities between pixel features and all stored class prototypes. When the number of classes becomes very large (e.g., thousands), this may increase computational cost and memory usage.

3) Scenario Scope: This work focuses primarily on the class-incremental setting. Whether the same framework generalizes effectively to other continual learning scenarios—such as task-incremental or domain-incremental learning—remains an open question.

D. Future Work

Based on our findings and the identified limitations, several future research directions appear promising:

1) More Advanced Prototype Strategies: Lightweight yet more adaptive prototype construction and update mechanisms could be explored. Examples include feature-space augmentation (e.g., controlled perturbations, interpolation, or hard-negative mining) and attention-inspired update rules that better track feature distribution shifts while remaining computationally efficient.

2) Hierarchical Prototype Memory: For datasets with hierarchical class structures (e.g., "wall → brick_wall → tile_wall" in ADE20K), hierarchical prototype representations could capture semantic relationships at multiple levels of granularity, potentially improving robustness and interpretability.

3) Integration with Other Continual Learning Techniques: Combining PMCD with parameter isolation strategies, sparse training methods, or modular architectures may further enhance resistance to forgetting and improve scalability.

4) Extension to Broader Tasks: Extending the PMCD framework to other dense prediction and multimodal tasks—such as continual object detection, instance segmentation, or cross-modal vision–language learning—would provide a stronger test of its generality and robustness.

In summary, PMCD introduces a structurally grounded approach to exemplar-free continual semantic segmentation. By preserving not only predictions but also the geometry of the representation space, it offers a robust and privacy-friendly pathway toward lifelong visual learning systems.

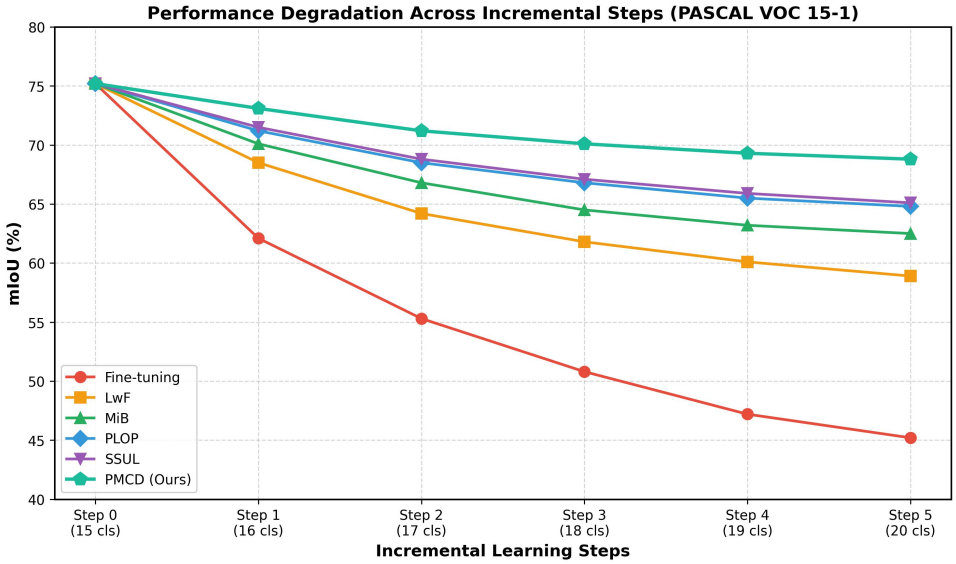

Fig. 6. Performance Degradation Across Incremental Steps on PASCAL VOC 15-1.

VI. Conclusion

In this work, we focused on one of the central challenges in continual semantic segmentation: how to effectively mitigate catastrophic forgetting without relying on access to historical data. To address this issue, we introduced a novel exemplar-free framework named Prototype Memory and Contrastive Distillation (PMCD). The proposed method incorporates a dynamically updated Prototype Memory Module to compactly encode and preserve the essential knowledge of previously learned classes. In addition, a carefully designed contrastive distillation mechanism explicitly enforces intra-class compactness and inter-class separability within the feature space. By shifting from passive logit alignment to active preservation of feature structure, PMCD offers a more principled and effective strategy for combating both knowledge forgetting and semantic drift.

Extensive experiments on benchmark datasets, including PASCAL VOC 2012 and ADE20K, demonstrate that PMCD consistently outperforms existing state-of-the-art approaches under various class-incremental learning settings. The results show that our method achieves a strong balance between retaining prior knowledge and acquiring new concepts, significantly reducing forgetting while maintaining high segmentation accuracy.

Beyond performance improvements, this research contributes a practical and privacy-friendly solution for continual semantic segmentation. Because PMCD does not require storing historical samples, it is particularly well suited for deployment in privacy-sensitive domains such as medical image analysis. More broadly, we believe that this prototype-based, metric-driven knowledge preservation paradigm provides a promising direction for future advances in continual learning and may inspire further research into structure-aware, exemplar-free learning systems.

References

[1] Zhou, T., Wang, W., Konukoglu, E., & Van Gool, L. (2022). Rethinking semantic segmentation: A prototype view. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4939 – 4949). https://doi.org/10.1109/CVPR52688.2022.00488

[2] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. Neural Networks, 113, 54–71. https://doi.org/10.1016/j.neunet.2019.01.012

[3] McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In R. G. M. Morris (Ed.), Psychology of Learning and Motivation (Vol. 24, pp. 109 – 165). Academic Press. https://doi.org/10.1016/S0079-7421(08)60536-6

[4] De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Tuytelaars, T., & Snoek, C. G. M. (2021). A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7), 3366 – 3385. https://doi.org/10.1109/TPAMI.2021.3057446

[5] Yuan, B., & Zhao, D. (2024). A Survey on Continual Semantic Segmentation: Theory, Challenge, Method and Application. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(12), 10891–10910. https://doi.org/10.1109/TPAMI.2024.3452013

[6] Michieli, U., & Zanuttigh, P. (2019). Incremental learning techniques for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. https://doi.org/10.1109/ICCVW.2019.00293

[7] Gonzalez, C., et al. (2025). What is Wrong with Continual Learning in Medical Image Segmentation? ACM Transactions on Computing for Healthcare. https://doi.org/10.1145/3706598.3713716

[8] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv Preprint arXiv:1503.02531. https://doi.org/10.48550/arXiv.1503.02531

[9] Douillard, A., et al. (2021). Plop: Learning without Forgetting for Continual Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR46437.2021.00444

[10] Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: Incremental classifier and representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2001 – 2010). https://doi.org/10.1109/CVPR.2017.587

[11] Shen, C., Liu, Y., Chen, B., Tao, X., Huangfu, Y., & Wang, D. (2025). Decoupling incremental classifier and representation learning based continual learning machinery fault diagnosis framework under long-tailed distribution. Chinese Journal of Mechanical Engineering, 100031. https://doi.org/10.1016/j.cjme.2025.100031

[12] Shang, C., Li, H., Meng, F., Wu, Q., Qiu, H., & Wang, L. (2023). Incrementer: Transformer for class-incremental semantic segmentation with knowledge distillation focusing on old class. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7214 – 7224). https://doi.org/10.1109/CVPR52729.2023.00696

[13] Li, Z., & Hoiem, D. (2017). Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(12), 2935–2947. https://doi.org/10.1109/TPAMI.2017.2773081

[14] Cermelli, F., et al. (2020). Modeling the background for incremental learning in semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR42600.2020.01193

[15] Wu, J., Jiang, Y., Yan, B., Lu, H., Yuan, Z., & Luo, P. (2023). Exploring transformers for open-world instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6611 – 6621). https://doi.org/10.1109/ICCV51070.2023.00608

[16] Rosch, E. (1975). Cognitive representations of semantic categories. Journal of Experimental Psychology: General, 104(3), 192 – 233. https://doi.org/10.1037/0096-3445.104.3.192

[17] Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical networks for few-shot learning. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1703.05175

[18] Zhou, T., & Wang, W. (2024). Prototype-based semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(10), 6858 – 6872. https://doi.org/10.1109/TPAMI.2024.3397956

[19] Zhu, F., et al. (2021). Prototype augmentation and self-supervision for incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR46437.2021.00433

[20] Tian, Y., Krishnan, D., & Isola, P. (2019). Contrastive representation distillation. arXiv Preprint arXiv:1910.10699. https://doi.org/10.48550/arXiv.1910.10699

[21] Oord, A. v. d., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv Preprint arXiv:1807.03748. https://doi.org/10.48550/arXiv.1807.03748

[22] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 801 – 818). Springer. https://doi.org/10.1007/978-3-030-01234-2_48

[23] Shafiq, M., & Gu, Z. (2022). Deep residual learning for image recognition: A survey. Applied Sciences, 12(18), 8972. https://doi.org/10.3390/app12188972

ACKNOWLEDGEMENTS

We would like to thank all participants involved in the experiments for their time and feedback. We also thank the anonymous reviewers for their constructive comments and suggestions. In addition, we appreciate the support from our colleagues and collaborators who provided helpful discussions, implementation advice, and assistance with experimental evaluation.

FUNDING

None.

AVAILABILITY OF DATA

Not applicable.

AUTHOR CONTRIBUTIONS

Hongzhong Bai contributed to the conceptualization, methodology, and manuscript drafting. Zhaoxiong Wen contributed to software implementation, experimental design, data analysis, and manuscript revision. Wenlong Xie contributed to experimental evaluation, validation of results, and editing for clarity and presentation. All authors reviewed and approved the final manuscript.

COMPETING INTERESTS

The authors declare no competing interests.