Songze Li[1], Tonghua Su[1], Xuyao Zhang[1], and Zhongjie Wang[1]

[1]Affiliation not available

July 21, 2024

Continual Learning with Knowledge Distillation: A Survey

Songze Li, Tonghua Su *Member, IEEE*, Xuyao Zhang *Senior Member, IEEE*, Zhongjie Wang *Member, IEEE*

***Abstract*—The foremost challenge in continual learning is to mitigate catastrophic forgetting, allowing a model to retain knowledge of previous tasks while learning new tasks. Knowledge distillation, a form of regularization, has gained significant attention for its ability to maintain a model's performance on previous tasks by mimicking the outputs of earlier models during the learning of new tasks, thus reducing forgetting. This paper offers a comprehensive survey of continual learning methods employing knowledge distillation within the realm of image classification. We provide a detailed analysis of how knowledge distillation is utilized in continual learning methods, categorizing its application into three distinct paradigms. Besides, we classify these methods based on the type of knowledge source used, and thoroughly examine how knowledge distillation consolidates memory in continual learning from the perspective of loss functions. In addition, we have conducted extensive experiments on CIFAR-100, TinyImageNet, and ImageNet-100 across ten knowledge distillation-integrated continual learning methods to analyse the role of knowledge distillation in continual learning, and we have further discussed its effectiveness in other continual learning tasks. Our extensive experimental evidence demonstrates that knowledge distillation plays a crucial role in mitigating forgetting in continual learning and substantiates that, when used with data replay, classification bias adversely affects the effectiveness of knowledge distillation, whereas employing a separated softmax loss can significantly enhance its efficacy.**

***Index Terms*—Continual Learning, Incremental Learning, Knowledge Distillation, Catastrophic Forgetting**

I. Introduction

Continual learning, an emergent field that addresses the need for adaptable intelligence, has garnered substantial interest recently. Standard deep learning methods learn a static data distribution from an established dataset, effectively targeting particular applications [1]. However, these methods encounter difficulties with data outside the distribution. Continual learning endeavors to enable models to assimilate new knowledge while preserving previous learning within dynamic data environments [2], [3]. This domain emphasizes the lifelong assimilation and refinement of knowledge across the lifespan of the model, analogous to human learning processes. In the literature, continual learning may also be called lifelong learning [4], [5], [6], incremental learning [7], [8], [9], or sequential learning [10], [11]. Continual learning's significance stems from its capacity to facilitate the progressive acquisition of knowledge from a continuous influx of new data, thereby precluding the necessity for complete model retraining.

Songze Li, Tonghua Su, and Zhongjie Wang are with the Faculty of Computing, Harbin Institute of Technology, Harbin, China (e-mail: lisongze@stu.hit.edu.cn, thsu@hit.edu.cn, rainy@hit.edu.cn)

Xuyao Zhang is with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation of Chinese Academy of Sciences, Beijing, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China (e-mail: xyz@nlpr.ia.ac.cn)

Manuscript received XXXXXX; XXXXXXXXX. Corresponding author: Zhongjie Wang (email: rainy@hit.edu.cn).

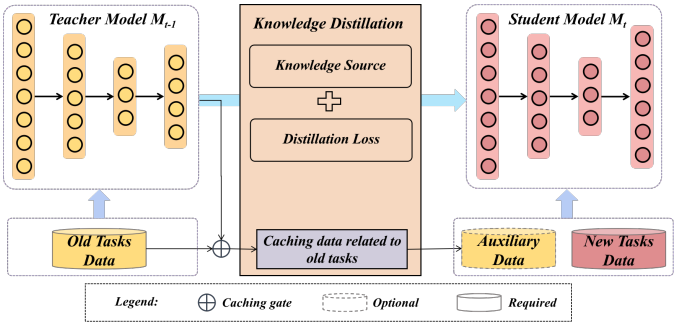

Fig. 1: Knowledge distillation used in continual learning framework. In continual learning, knowledge distillation follows a Teacher-Student paradigm in which the old tasks model M_{t-1} serves as a teacher model and the new tasks model M_t acts as a student model. Knowledge transfers from teacher model to student model to preserve old tasks knowledge while learning new tasks. Many methods are devised in light of knowledge sources and distillation losses. Moreover, to enhance memory retention, auxiliary data related to old tasks are cached for replaying.

In contrast to traditional joint training, where the model has access to the entire dataset, continual learning models are often challenged by the phenomenon of **catastrophic forgetting** [10], [12] due to their inability to utilize historical data. This leads to a failure in retaining previously acquired knowledge upon the assimilation of new information. To tackle this issue, models must be designed to mitigate the loss of older knowledge while also facilitating the integration of new knowledge during the continual learning cycle. The capacity of a model to assimilate new knowledge and simultaneously preserve existing information is encapsulated by the **stability-plasticity dilemma** [13], [14]. The overarching aim of continual learning is to achieve an equilibrium between stability and plasticity—enabling the model to maintain critical knowledge and effectively incorporate emergent information.

A variety of strategies have been proposed to curb catastrophic forgetting within the realm of continual learning. Knowledge distillation (KD), serving as regularization-based method [8], [15], has progressively become a standard technique for alleviating the forgetting issue in continual learning by prompting the new task model to emulate the output of the old task model. Within the continual learning framework,

the previously trained model acts as a "teacher" that progressively relays its expertise to the "student" model which learns new tasks [16], [17]. This process ensures the retention of learned information while facilitating the acquisition of novel tasks. KD for continual learning is often conceptualized as a form of self-distillation [18], [19], wherein the teacher and student models share identical architectures except for their classification layers. The adoption of KD in conjunction with other methodologies, such as data replay, has seen a surge in popularity [20], [15]. Fig. 1 depicts the application of KD within continual learning framework.

Although KD has steadily become a prevalent method for countering catastrophic forgetting, a thorough exploration is still needed to understand its integration into continual learning practices and its effectiveness in overcoming forgetting. Most contemporary surveys on continual learning primarily investigate the field from various methodological categorizations in image classification domain and other application domains. There is a notable scarcity of reviews analyzing the field through the lens of specific techniques (such as KD) aimed at mitigating the issue of forgetting in continual learning. For example, [21] is the first survey to systematically categorize continual learning methods into three types: replay-based methods, regularization-based methods, and parameter isolation-based methods. [7] summarizes the types of continual learning scenarios, defining three types: task-incremental learning, class-incremental learning, and domain-incremental learning. [22] defines six essential properties of continual learning algorithms. [23] focuses on online continual learning and summarizes the evaluation metrics in continual learning. [24] and [25] discuss continual learning from a biological perspective. [8] and [9] conduct a targeted survey of class-incremental continual learning methods with the growing attention to class-incremental learning scenarios. [15] provides an synthesis analysis of continual learning from theory, representative methods, and applications perspective. Beyond image classification task, some other surveys focus on continual learning in other application domains, such as robotics [26], natural language processing [27], [28] and neural recommender systems[29].

In this survey, we undertake a scrutiny of continual learning methods that implement KD, primarily within the realm of image classification tasks. We conduct a detailed analysis of how knowledge distillation (KD) is utilized in continual learning methods, broadly categorizing its application into three distinct paradigms: KD as Regularization, Combination of KD with Data Replay, and Combination of KD with Feature Replay. Subsequently, we classify these methods based on the source of knowledge used in the distillation process, identifying three primary levels: logits-level, feature-level, and data-level. We then conduct an in-depth analysis from the perspective of distillation loss functions to understand how KD consolidates memory in continual learning. To investigate the impact of KD on mitigating forgetting, we select ten continual learning methods that incorporate KD and conducted extensive experiments on datasets such as CIFAR-100, TinyImageNet, and ImageNet-100, thoroughly analyzing the role of KD in various continual learning scenarios. Additionally, we substantiate separated softmax classification loss can enhance the effectiveness of KD in mitigating forgetting when used in conjunction with data replay. Overall, the principal contributions of this survey are delineated as follows:

- We present a comprehensive investigation of KD-integrated continual learning methods which mainly focus on image classification tasks. To our knowledge, this is the first systematic review in this area.
- We introduce a novel taxonomy to categorize the KD-integrated continual learning methods from two aspects: the application paradigms of KD in continual learning and the distilled knowledge sources. Our detailed analysis delves into distillation loss for combating forgetting.
- We conduct extensive experiments with ten KD-integrated continual learning methods on widely-adopted datasets across diverse continual learning scenarios to show the role of KD in alleviating forgetting.
- We substantiate that classification bias can impair the performance of KD and separated softmax loss can enhance the effectiveness of KD in mitigating forgetting when used in conjunction with data replay.

This survey is structured as follows: Sec. II provides background and problem formulation on KD and continual learning. Sec. III analyses the application paradigms of KD applied in continual learning. Sec. IV categorizes KD-integrated continual learning methods by the distilled knowledge source and examine how KD consolidates memory from the perspective of KD loss functions. Sec. V details the experimental methodology and analyzes KD's effects on continual learning extensively. Sec. VI discusses the effectiveness of KD to mitigate forgetting in other vision tasks. Finally, Sec. VII outlines trends for future research in continual learning with KD and Sec. VIII concludes our survey.

II. Preliminaries and Problem Formulation

In this section, we first elucidate the concepts of KD and continual learning, followed by a detailed description of various continual learning protocols. Subsequently, we formalize the integration of KD within the continual learning framework.

A. Knowledge Distillation

The fundamental aim of KD is to transfer knowledge [30]. It follows a teacher-student schema [16], wherein the student model is trained to emulate the outputs of the well-trained teacher model. The concept of KD is introduced by [31], where knowledge transfer from teacher to student model is achieved by minimizing the Kullback–Leibler divergence [32] between their outputs. Let Ter represents the teacher model and Stu the student model. The output logits of the teacher model for input x are denoted as $z_{Ter}(x)$, and those of the student model as $z_{Stu}(x)$. The soft targets for distillation are obtained by applying a softmax function with temperature τ:

$$P_{Ter}(x) = \text{softmax}\left(z_{Ter}(x)/\tau\right), \tag{1}$$

$$P_{Stu}(x) = \text{softmax}\left(z_{Stu}(x)/\tau\right). \tag{2}$$

(a) Class, Task and Domain incremental learning

(b) Balanced, Few-Shot and Long-Tailed CIL

(c) Offline and Online CIL

Fig. 2: Continual Learning Protocols. The number of new classes grows over time in CIL. Inference is performed without task IDs, whereas TIL requires task IDs. In DIL, the number of classes is fixed but their data distributions change. Balanced CIL is assumed where each class has an equal amount of training data. In Few-Shot CIL, N samples are available for the first task, while only K samples per class are available in subsequent tasks, where N represents a relatively large quantity and K represents a relatively small quantity. In Long-Tailed CIL, the data is imbalanced with differing amounts per class. The majority of continual learning settings are Offline CIL, allowing full passes over the current task data. Online CIL permits only a single pass over data as it arrives.

The distillation loss L_{KD} is defined as the Kullback-Leibler (KL) divergence between the teacher and student soft targets:

$$L_{KD} = KL(P_{Ter}(x) \parallel P_{Stu}(x)). \quad (3)$$

This loss is minimized during training to ensure the student model emulates the teacher model.

Initially, the main application of KD is model compression [33], [34], addressing the increased computational and storage demands due to the growth in deep learning model size, which limits suitability for real-time processing and deployment on resource-constrained systems. Except for model compression, the application of KD extends to various domains such as privileged learning [35], [36], mutual learning [37], [38], assistant teaching [39], [40], self-learning [41], [42], protective measures against adversarial attacks [43], [44], and notably, continual learning [45], [46].

B. Continual Learning

Continual learning aims to learn from a continuous data stream with changing distributions, retaining past knowledge even as new information is acquired. Unlike static train-and-deploy models, continual learning models are designed to grow and incorporate new skills and knowledge over time. However, these models frequently face catastrophic forgetting, where the introduction of new tasks without access to old task data causes loss of previously learned information.

Approaches to mitigating catastrophic forgetting in continual learning can be broadly classified into three main strategies. The first strategy is usually known as regularization-based methods [47], [48], [49], which employs regularization techniques to steer model training toward a global optimum that accommodates all tasks. Another strategy is parameter isolation-based or architecture-based, where distinct parameters are allocated for each task during continual learning process. These approaches can be further divided into fixed [50], [51], [52] and dynamic [53], [54], [55] architectural models based on changes in network size throughout the learning process. Finally, certain methods mitigate forgetting by caching a small subset of data from previous tasks and replaying this data during learning new tasks. This technique is commonly referred to as replay-based methods [56], [57], [58].

C. Continual Learning Protocols

This subsection offers an in-depth look at experimental protocols in continual learning, analyzing them from three perspectives. First, we categorize continual learning into Class Incremental Learning (CIL) [59], [60], [61], [62], Task Incremental Learning (TIL) [7], [21], and Domain Incremental Learning (DIL) [63], [64], [65], considering the updates to the model's classification head and the need for task indicators during training and inference. Among these, CIL is currently one of the most prevalent protocols within the field of continual learning research. Based on CIL, we dichotomize continual learning into offline and online [66], [67], [68] CIL depending on the number of times a learning model accesses the data stream during training. Additionally, we classify the continual learning based on the count of training samples across classes, resulting in balanced, few-shot [69], [70], [71], and long-tailed CIL [72], [73]. The characteristics of the various continual learning protocols are illustrated in Fig. 2.

Class, Task and Domain Incremental Learning. Class CIL is the most prevalent in continual learning, involving a series of tasks with unique labels to enable classification across all learned classes without task identifiers. The model's classification heads grow with each new task. TIL resembles CIL but uses a task indicator during training and inference, allowing differentiation within the scope of a given task thereby simplifying class determination compared to CIL. DIL maintains a constant number of classification heads, facing the challenge of classifying samples from the same classes that exhibit varying distributions over time, delivered sequentially.

Offline and Online CIL. When not explicitly specifying "Online" CIL the term typically defaults to "Offline" CIL. Offline CIL allows models to iterate over the current task's data stream until they reach convergence. This setup permits repeated training cycles for performance enhancement. In contrast, Online CIL restricts models to a one-time pass through incoming data, simulating real-world conditions where data is ephemeral and can only be preserved through physical capture. This single-pass restriction heightens the difficulty as models must quickly learn and integrate new information without revisiting the data.

Balanced, Few-Shot and Long-Tailed CIL. Existing methods in the field of continual learning often operate under the assumption that the data associated with incoming tasks is uniformly distributed, with each task providing an equivalent volume of training samples. We refer to this protocol as balanced CIL which is atypical in real-world settings. Few-Shot CIL (FSCIL) [69] blends few-shot learning with continual learning. In FSCIL, the model begins with extensive training data for the initial task, while subsequent tasks are introduced with only a few samples each, significantly fewer than the first task's dataset. Long-Tailed CIL (LT-CIL) [72] acknowledges the complexity and imbalance of real-life data, addressing disparate data distributions across tasks. Tasks are sequenced by different sample volume, reflecting the long-tailed distribution found in real-world datasets, thus mirroring the diversity and imbalance typical in actual use cases.

D. Formulation of Continual Learning with KD

In our study, we focus on CIL protocol wherein the model is exposed to a continuous data stream characterized by a sequence of independent tasks which is defined as $D = \{D_1, \ldots, D_t, \ldots, D_T\}$, where t denotes the index of a given task and T represents the total number of tasks. The tasks are introduced in a sequential manner, and the model is expected to learn from the current task D_t without the opportunity to revisit data from preceding tasks. The continual learning model is conceptualized as $H = \{\theta, \phi\}$, which includes a feature extractor θ and a classifier ϕ. For each task indexed by t, the model encounters data $D_t = \{X_t, Y_t\}$, wherein X_t signifies the set of inputs and Y_t corresponds to the associated

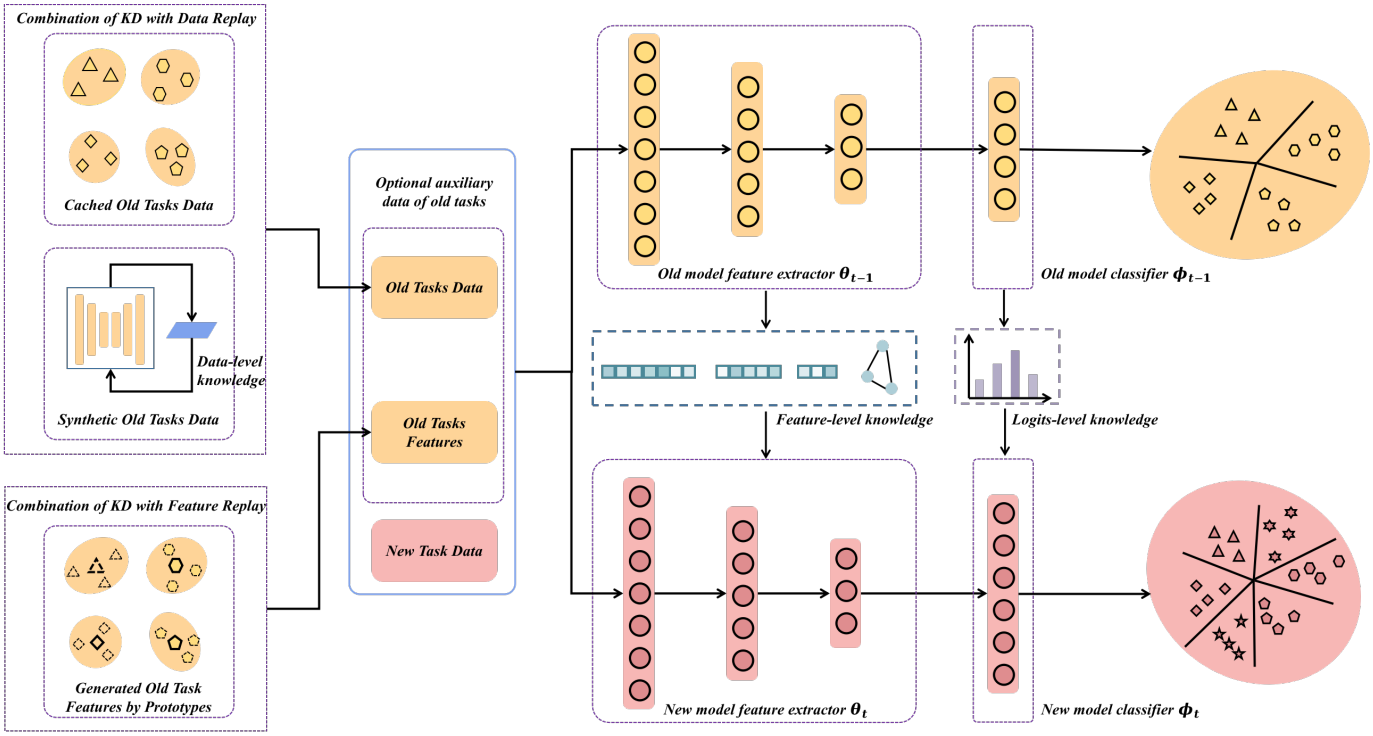

Fig. 3: The schematic structure for continual learning with KD. Knowledge transfers from old tasks model (yellow network with feature extractor θ_{t-1} and classifier ϕ_{t-1}) to new task model (red network with feature extractor θ_t and classifier ϕ_t). Knowledge comes from different levels such as Logits-level, Feature-level and Data-level. To support old knowledge retention while acquiring new skills, auxiliary data about previous tasks is provided, which might include cached or synthetic old tasks data, generated features by cached prototypes, or nothing.

labels, with each task comprising C_t classes. During the training phase on D_t, inputs X_t are processed through θ to feature vectors $f = \theta(X_t)$. Subsequently, these features f are fed to ϕ to generate the pre-softmax logits $z = \phi(f)$. A fundamental challenge in this setting is the proclivity of H to catastrophically forget prior knowledge when past samples are not retained.

To counteract catastrophic forgetting, a memory buffer M is usually used to retain a selection of samples from previous tasks, facilitating memory replay during new task learning. The objective of the continual learning can be modeled to minimize the loss over both current task data and memory buffer samples, formally defined as:

$$L_{CL} = \mathbb{E}_{(x,y)\sim D_t \cup M} \left[\mathcal{L}(\phi(\theta(x)), y) \right], \tag{4}$$

where $\mathcal{L}$ is the loss function measuring the discrepancy between the predicted labels and the true labels, typically using cross-entropy loss.

Additionally, KD is often integrated into the continual learning framework as a regularization term to preserve knowledge from previous tasks. Unlike compression-centric distillation, KD in continual learning adopts a self-distillation framework, where the teacher and student models share the same network architecture. In this context, the model trained on the previous task serves as the teacher model, transferring its knowledge to the new task model during the learning of new tasks, thereby preserving the memory of old tasks. The regularization term L_{KD} in continual learning context is defined as:

$$L_{KD} = \mathbb{E}_{(x,y)\sim D_t \cup M} \left[\mathcal{KD}\left(H_{t-1}(x), H_t(x)\right) \right], \tag{5}$$

where $H_{t-1}(x)$ and $H_t(x)$ represent the knowledge from the old task model and the new task model, respectively. The function $\mathcal{KD}$ denotes different types of distillation losses, which will be discussed in detail in Sec. IV. This term ensures that the student model retains the knowledge from previous tasks while learning new ones. Thus, the final objective function combining with KD is:

$$L_{CL} = \mathbb{E}_{(x,y)\sim D_t \cup M} \left[\mathcal{L}(\phi(\theta(x)), y) \right] + L_{KD}. \tag{6}$$

III. Paradigms of Continual Learning with KD

In this section, we conduct an in-depth analysis of the application paradigms of KD in continual learning methods. We discovered that KD is initially introduced as a standalone regularization technique to prevent forgetting in continual learning. However, as continual learning methods evolve, KD has been increasingly integrated and complemented with various other techniques such as data replay and feature replay. Based on this, we categorize the application of KD in continual learning into three paradigms: KD as Regularization, Combination of KD with Data Replay, Combination of KD with Feature Replay. Fig. 3 offers a detailed diagram of our classification framework.

A. KD as Regularization

Initially, KD was introduced into continual learning as an independent regularization technique, aiming to prevent forgetting previously learned knowledge by constraining model parameter changes. This type of method employs KD as the core mechanism to combat forgetting, with its performance heavily dependent on the effectiveness of the KD process.

LwF [45] is the pioneering approach that uses KD as a regularization term within continual learning by distilling the responses of new task data on the old model, thereby preserving the memory of the old task. Similarly, LwM [74] leverages KD to suppress forgetting by maintaining the attention regions of the old task model and combines this with basic logits-level distillation. DMC [59] further emphasizes the role of KD by using a double distillation loss which distilling combined logits from new task expert models and old task models. BLD [75] proposes a batch level distillation in which the distillation data is collected from a warm-up stage and distillation step happens in a joint learning stage. TPCIL [76] captures knowledge as an Elastic Hebbian Graph (EHG) in feature space, with nodes representing class prototypes and edges for inter-class cosine distances. TPCIL uses a topology-preserving loss to penalize structural changes to the EHG during incremental learning. Here, KD is pivotal in maintaining the integrity of the EHG's structure, ensuring that learned knowledge is not forgotten. PRD [77] continually learns the feature extractor with supervised contrastive learning which is demonstrated to be less prone to forgetting and evolves old tasks prototypes to a new feature space with a relation distillation loss between old tasks prototypes and new tasks samples. Methods in the "KD as Regularization" paradigm use KD as the primary and sole approach to mitigate forgetting. The idea of these approach is straightforward. However, its overall ability to prevent forgetting is relatively poor, which usually achieves lower performance.

B. Combination of KD with Data Replay

Recently KD is frequently combined with data replay techniques to enhance memory retention from both data and model perspectives. Data replay methods need an additional buffer to cache additional samples from previous tasks to approximate their distribution, continuously replaying these samples during continual learning to achieve sustained memory retention. Integrating KD with data replay methods further enhances a model's memory retention capability.

iCaRL [46] is the first method to combine KD and data replay. After that, many methods that combine KD and data replay treat data replay is as a fundamental technique to combat forgetting and explore various types of distillation techniques to further enhance the memory retention of old tasks. EEIL [78] introduces a task-wise distillation loss to avoid forgetting. D+R [79] distills knowledge from both the old task model and an intermediate expert model for new task. GD [80] also trains an intermediate new task expert and uses global distillation with unlabeled data to distill from this expert, the old model, and their combined knowledge. ILOS [66] proposes a modified cross-distillation method to deal with online scenarios. PODNet [81] introduces a spatial-based feature distillation that maintains valuable features throughout continual learning. AFC [82] devises an importance-based feature distillation method that conserves critical features and allows more adaptability for less important ones during new task learning. Co2L [83] proposes an instance-wise relation distillation method that maintains memory by preserving local topological relationships between samples. GeoDL [84] projects the features onto low-dimensional manifold subspaces before distillation. COIL [61] proposes a bidirectional distillation loss which transfers knowledge forward and backward to help model fast adapting and prevent catastrophic forgetting. MBP [85] aims to maintain model behavior with an instance neighborhood-preserving loss to prevent changes in instance relationships, alongside a label priority-preserving loss to avoid shifts in class priority. DER++ [86] finds that distillation on logits trajectory in training process have a better effect on consolidating memory. XDER [60] improves on DER++ [86] by updating the distilled logits based on the future task data's influence on the logits, resulting in better performance. OCD-Net [87] uses online response distillation to counter teacher model bias, along with adaptive perception adjustments that enhance the teacher's response quality. In addition to directly using extra memory to store replay data from old tasks, some methods generate replay data through generative models [88], [89] or model inversion [90], [91], [92] techniques. These methods typically apply KD to the generated data to prevent the generative model from forgetting during the continual learning process, while also using basic KD techniques on logits or features to mitigate forgetting. For these methods, in addition to KD being an effective means of mitigating forgetting, the quality of the generated data also plays a crucial role in determining the overall effectiveness of forgetting mitigation.

Due to the severe data imbalance between the small amount of replay data and the new task data, which easily leads to classification bias, some methods, in addition to treating KD as a standard mechanism for memory retention, place greater emphasis on addressing the classification bias issue. For example, BiC [93] explicitly addresses the classifier bias problem by training class correction parameters on a balanced validation dataset, while RDICL [94] fixes this issue through a dynamic threshold moving algorithm. WA [95] proposes weight aligning to correct bias without requiring extra correction parameters, unlike BiC [93]. LUCIR [96] tackles the imbalanced classification problem by normalizing features and increasing inter-class separation. GD [80] mitigates bias by simulating multiple data feedings with scaled gradients during the balanced fine-tuning stage. SS-IL [97] uses separated softmax to alleviate bias issue. LVT [98] designs a dual classifier system, one for new task feature learning and another for integrating knowledge from all tasks in a balanced manner. DRC [99] alleviates the classifier bias problem by designing a dynamic residual classifier. In addition to classification bias issue, some other methods treat the combination of KD with data replay as a standard mechanism for memory retention, and pay more attention on other problems, such as the selection of replay data [100], [101] and incorporating architecture-

based approaches [102], [103], [104] to maintain memory.

C. Combination of KD with Feature Replay

In addition to combining KD with data replay, many methods integrate KD with feature replay to achieve continual learning without exemplars. Most methods of this paradigm utilize instance feature alignment in feature-level distillation which will be explained in Sec. IV to maintain the memory capabilities of the feature network, and employ various feature generation methods to produce features for replay, thereby ensuring the classifier's memory is preserved. GFR [105] stores the features of old tasks by training a generative model, which then generates features for replay during continual learning. PASS [106] defines class prototypes as the mean of feature space data and introduces Gaussian noise for augmentation during new class learning, preventing classification bias toward new data. IL2A [107], like PASS [106], represents old class distributions with mean-based prototypes but also includes distribution variance for feature space data augmentation to protect old class boundaries during new class learning. FRoST [108] also replays generative features drawn from a Gaussian distribution, which is defined by stored feature prototypes and variances of old class data. Different from PASS [106], SSRE [62] chooses to over-sample prototypes which is adopted in long-tail recognition [109] to generate replay features. FeTrIL [110] generates pseudo-features for past classes by applying a simple geometric translation to new class features, with the crucial condition that the feature extractor remains fixed as learned in the initial stage. PRAKA [111] generates replay features by randomly interpolating bidirectionally between the extracted new class features and the stored old class prototypes. Fusion [112] observes feature distribution drift with new class introduction, causing changes to prototypes. To gauge and correct prototype drift, it employs a DNN to parameterize either a Gaussian or variational model before classifier training. MEIL [113] combats forgetting by replaying cached old task features. With feature space drift after learning new tasks, it transfers these cached features to updated feature space via a feature adaptation module. Besides, MEIL uses both logits and feature distillation to maintain the old model's responses to new data during learning new tasks. Compared to methods in the "Combination of KD with Data Replay" paradigm, this paradigm does not require a large amount of extra memory to store original samples from old tasks. Instead, it only needs a small amount of memory to store feature information for each class. Additionally, feature replay helps reduce the classification bias problem caused by the imbalance between replay data and new task data.

IV. Knowledge Sources and KD Losses

In this section, we categorize KD-integrated continual learning methods according to knowledge sources into three types: logits-level, feature-level, and data-level (as shown in Fig. 4). We also explore how KD consolidates memory from the perspective of distillation loss under different types of knowledge sources.

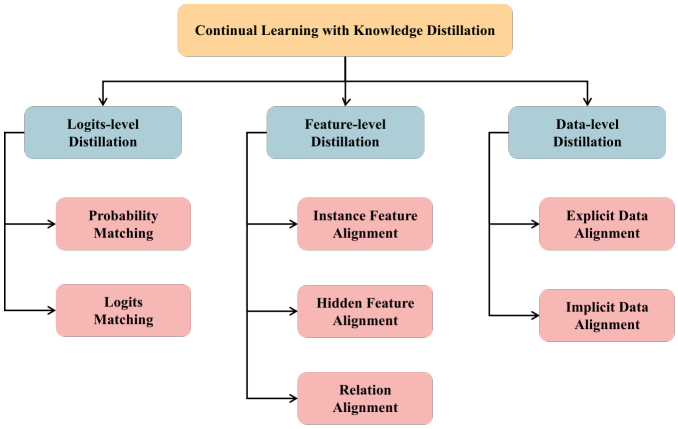

Fig. 4: Continual learning with KD from different knowledge sources.

A. Logits-level Distillation

Logits-level distillation primarily involves the student model assimilating knowledge by emulating the teacher model's final output logits. These outputs generally constitute two types: normalized classification probabilities through a normalization function (e.g., softmax) and the raw, unnormalized logits. Consequently, we categorize logits-level KD methods in continual learning into two subcategories: Probability Matching and Logits Matching. Probability Matching is common, with the student aiming to match the teacher's output probability distribution using loss functions like KL divergence or cross-entropy. In contrast, Logits Matching aims to synchronize the pre-softmax logit values, often using loss functions like L1 or L2 norms. Logits Matching imposes more stringent distillation constraints than Probability Matching. The left-hand section of Fig. 5 provides a conceptual illustration, detailing the various loss functions employed in logits-level distillation.

Probability Matching. LwF [45] is the first to introduce KD in continual learning. It encourages the output probability distribution of new task learner model to approximate the output probability distribution of old task learner model in continual learning progress. It uses a modified cross-entropy loss to achieve the purpose:

$$L_{KD} = -\sum_{i=1}^{C_{1:t-1}} s(\pi(\hat{z}_i))log s(\pi(z_i)), s(p_i) = \frac{p_i^{1/\tau}}{\sum_j p_j^{1/\tau}}, \quad (7)$$

where $\hat{z}_i$ is output logit of old model, z_i is output logit of new model, π is softmax function, $C_{1:t-1}$ is total number of classes for old tasks, s is a smooth function with temperature τ. This form of KD was subsequently widely adopted in several works [93], [95], [114], [100], [66]. However, instead of the separate smooth loss function, these methods directly incorporate temperature in the softmax to perform distillation, as shown below:

$$L_{KD} = -\sum_{i=1}^{C_{1:t-1}} \pi(\hat{z}_i/\tau)log\pi(z_i/\tau). \quad (8)$$

For simplicity and brevity, we hereafter denote the temperature-scaled softmax function as π by default in all subsequent formulas.

iCaRL [46] is the first method that introduce rehearsal memory into continual learning and combine KD with memory replay to alleviate catastrophic forgetting. iCaRL proposes a loss that inject classification into KD by interpreting network output logit as probability with a sigmoid function. It distills the old tasks knowledge with the post-sigmoid probability and classify new categories with a binary cross entropy classification loss:

$$L_{KD} = -(\sum_{i=1}^{C_{1:t-1}} \sigma(\hat{z}_i) log \sigma(z_i) + (1-\sigma(\hat{z}_i)) log(1-\sigma(z_i))), \tag{9}$$

where σ is $sigmoid$ function.

EEIL [78] devises an end-to-end continual learning paradigm with a cross-distilled loss which incorporate cross entropy and distillation loss. It consolidates old task knowledge with individual probability matching for each old task which we call it task-wise distillation (TKD):

$$L_{KD} = -\sum_{i=1}^{t-1}\sum_{j=1}^{C_i} \pi(\hat{z_{ij}}) log \pi(z_{ij}), \tag{10}$$

where t is index of current task, C_i is number of classes for each task. SS-IL [97] proves the TKD is powerful when combined with Separated-Softmax classification layer.

D+R [79] combines distillation and retrospection to achieve a better balance between preservation on old knowledge and adaptation on new knowledge during continual learning. It distills knowledge from two teacher models, one is an intermediate expert model which is learned only from current new task data to adapt the target model to new task and meanwhile consolidates old task knowledge by KD on previous model.

$$L_{KD} = -\sum_{i=1}^{C_{1:t-1}} \pi(\hat{z}_i) log \pi(z_i) - \sum_{i=C_{1:t-1}+1}^{C_{1:t}} \pi(q_i) log \pi(z_i), \tag{11}$$

where $C_{1:t}$ is number of classes until current task, and q_i is the response of expert model.

GD [80] designs a global distillation loss to maintain previous tasks knowledge by leveraging a large stream of unlabeled wild data which is easily obtainable and helpful to avoid overfitting to recent task. The global distillation loss comprises three parts: distillation from old tasks model M_{t-1}, distillation from current task expert model M_{expert} and distillation of ensemble knowledge from M_{t-1} and M_{expert} to complete the missing knowledge with external wild data.

MBP [85] alleviates forgetting with a relaxed probability matching scheme that does not strictly match the post-softmax output distributions between the new and old task models. Instead, it matches the label priority vectors after sorting according to the probabilities vector from old and new task model. This helps to retains the semantic relevance between different classes learned by the model.

BLD [75] proposes a Batch-Level Distillation loss to balance stability and plasticity during online continual learning under extremely limited storage conditions. It computes the classification probabilities of old task in the warm-up training stage and prevent catastrophic forgetting by distill the cached classification probabilities for each old task independently at a mini-batch level in the joint learning stage.

Logits Matching. DER++ [86] distills past experience by matching the network pre-softmax output logits sampled over the optimization trajectory rather than final optimized network's output logits. It empirically proves that distillation on training trajectory can converge to a flatter loss landscape and achieve better model calibration. The knowledge is inherited with the Euclidean distance loss instead of the KL divergence to eliminate the information loss from squashing function like softmax:

$$L_{KD} = \|\hat{z} - z\|_2. \tag{12}$$

XDER [60] indicates there are two pitfalls in DER++ [86]. The stored logits trajectories in DER++ have a blind spot for future task because they are learned from already observed data without any prognosis of future task data. XDER keeps the future part of pre-softmax logits update-to-date after learning new tasks with secondary information, and distills knowledge from the updated logits with the same loss function as used by DER++.

DMC [59] introduces a Deep Model Consolidation paradigm, which compacts knowledge from new task expert model and old task model with a double distillation loss. The compacted model is learned by distilling the concatenated knowledge with the help of auxiliary unlabeled data. The double distillation loss is defined as follows:

$$L_{KD} = \frac{1}{C_{1:t}} \sum_{i=1}^{C_{1:t}} (z_i - \tilde{z}_i)^2, \tag{13}$$

$$\tilde{z}_i = \begin{cases} \bar{z}_i - \frac{1}{C_{1:t-1}} \sum_{j=1}^{C_{1:t-1}} \bar{z}_j, & \text{if } 1 \le j \le C_{1:t-1} \\ \bar{z}_i - \frac{1}{C_t} \sum_{j=C_{1:t-1}+1}^{C_{1:t}} \bar{z}_j, & \text{if } C_{1:t-1} < j \le C_{1:t} \end{cases}, \tag{14}$$

where z_i represents the logits output of target compact model and $\tilde{z}_i$ is the concatenated logits from the old task model and the new task expert model. The term $\bar{z}_i$ refers to the logits output from the old task model if $1 \le j \le C_{1:t-1}$, or from the new task expert mode if $C_{1:t-1} < j \le C_{1:t}$.

R-DFCIL [91] introduces Hard Knowledge Distillation (HKD) and applies it to the output logits of synthesized old-task data obtained through model inversion [115] to consolidate the distribution of old-task data. R-DFCIL found that using KL divergence with temperature-scaled softmax functions to align the outputs of synthesized data on both the old and new models lacked sufficient constraint power. Hence, it applies an L1 loss constraint on the pre-softmax logits to enhance the consolidation of old knowledge:

$$L_{KD} = \|\hat{z} - z\|_1. \tag{15}$$

OCD-Net [87] devises online and adaptive pre-softmax logits distillation to counteract inherent response bias from the teacher model during offline KD, where the static teacher model can develop classification bias due to class imbalances in training data. OCD-Net updates the teacher model with the momentum update technique [116], ensuring the teacher model's responses are up-to-date. Additionally, OCD-Net employs adaptive perception to modulate the distilled

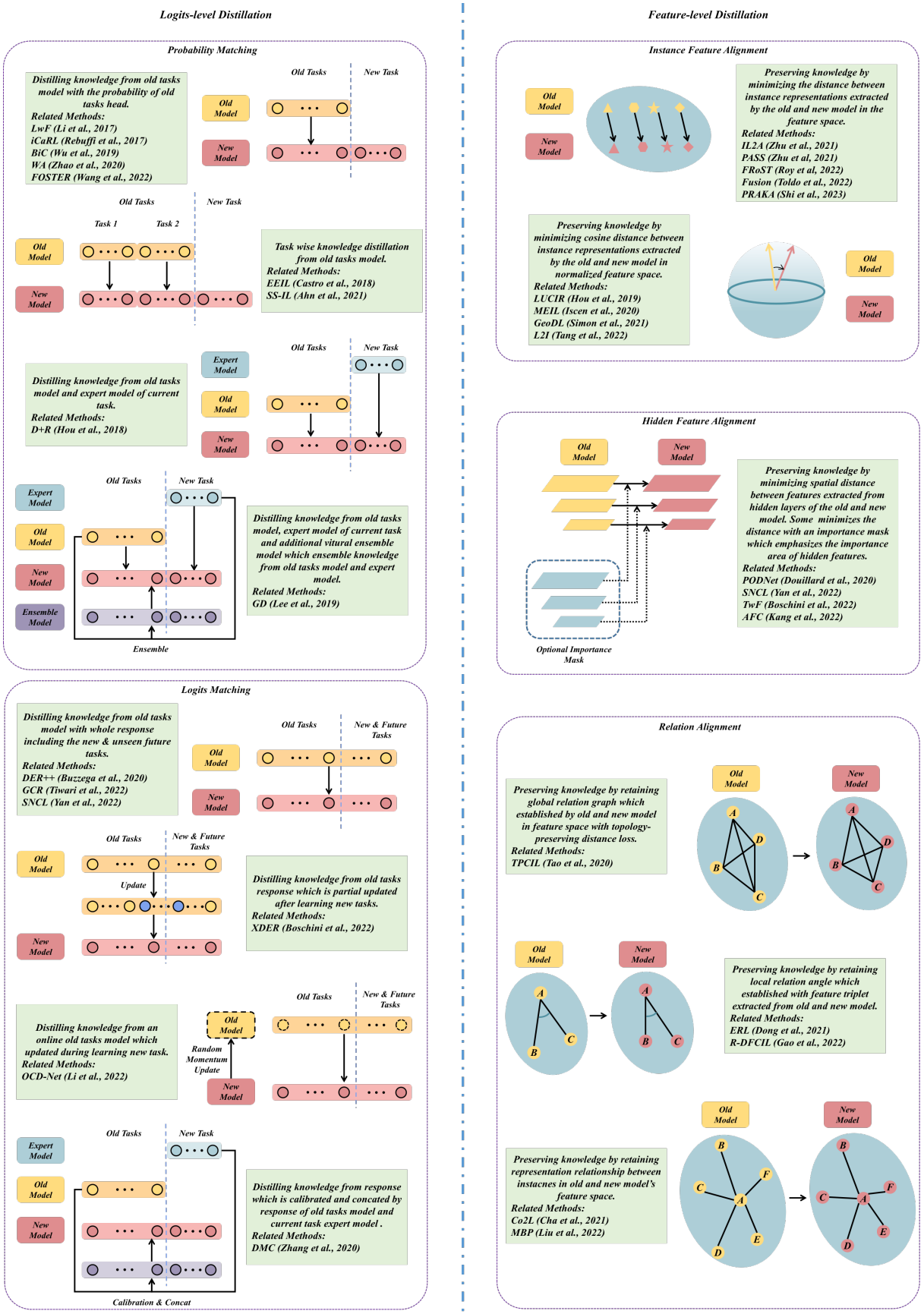

Fig. 5: Different forms of KD loss at logits-level and feature-level. **Left:** Logits-level distillation losses. **Right:** Feature-level distillation losses. The directed solid line is the knowledge transfer direction in continual learning.

logits, enhancing the student model's ability to learn high-quality responses from the teacher. The following is the online response distillation with adaptive perception:

$$L_{KD} = \omega \| \hat{z} - z \|_2, \tag{16}$$

$$\omega = \frac{exp(\hat{z_{gt}}/\tau)}{\sum_{i=1}^{C_{1:t}} exp(\hat{z_i}/\tau)}, \tag{17}$$

where ω is the quality score which is obtained by 17 where $\hat{z_{gt}}$ represents the response corresponding to the ground truth class of the input sample.

B. Feature-level Distillation

Feature-level distillation seeks to impart knowledge derived from the internal representations produced during a network's feature extraction phase. This type of methods can be divided into three subclasses based on the characteristics and location of the features within the network: Instance Feature Alignment, Hidden Feature Alignment, and Relation Alignment. Instance Feature Alignment targets the distillation of features from individual inputs presented to the classifier, which are usually converted into one-dimensional vectors. Hidden Feature Alignment focuses on the distillation of the intermediate layers' features within the feature extractor, retaining spatial information inherent to the network's structure. Relation Alignment, by contrast, aims to distill the intricate local or global relational dynamics among the features pertaining to multiple instances or prototype representations. The right half of Fig. 5 presents a schematic representation that delineates the loss mechanisms inherent in diverse feature-level distillation.

Instance Feature Alignment. LUCIR [96] mitigates the adverse effects of catastrophic forgetting in continual learning with a distillation loss on normalized instance feature. It uses cosine distance to measure similarity between two normalized features and prevents features from changing and rotating dramatically with a constraint on feature cosine distance:

$$L_{KD} = 1 - \langle \frac{\hat{f}}{\|\hat{f}\|}, \frac{f}{\|f\|} \rangle, \tag{18}$$

where $\langle \hat{f}, f \rangle = \hat{f}^T f$, $\hat{f}$ and f are features from old task model and current model. The feature cosine angle distillation is also adopted by [113], [117], [84].

GFR [105] restricts the change in generated features used for feature replay to prevent catastrophic forgetting. It trains a generator to memory features learned and replays them during continual learning to address imbalance problem on classifier. The feature extractor preserves old task knowledge with a feature distillation via L2 loss:

$$L_{KD} = \| \hat{f} - f \|_2. \tag{19}$$

[107], [106], [112], [108], [118], [62], [110], [111] follow it to use L2 feature distillation loss.

Hidden Feature Alignment. PODNet [81] fights catastrophic forgetting with a spatial-based distillation loss which applied on intermediate features and final instance feature. The PODNet [81] distills pooled intermediate feature after convolution layers and empirically finds that a spatial-based distillation pattern in height and width directions as Eq. 20 can achieve a better stability and plasticity trade-off. For the final output feature of feature extractor, it distills knowledge as Eq. 19.

$$\begin{aligned} L_{KD} = &\sum_{c=1}^{C} \sum_{h=1}^{H} \| \sum_{w=1}^{W} \hat{f}_{l,c,w,h} - \sum_{w=1}^{W} f_{l,c,w,h} \|_2 \\ &+ \sum_{c=1}^{C} \sum_{w=1}^{W} \| \sum_{h=1}^{H} \hat{f}_{l,c,w,h} - \sum_{h=1}^{H} f_{l,c,w,h} \|_2, \end{aligned} \tag{20}$$

where c, h, w are index for intermediate feature in channel, height, width axes, l is index of intermediate layer.

TwF [119] employs a weighted feature distillation on model's intermediate layers to address the issue of underutilized pre-trained models in continual learning. Directly distilling these features through Euclidean distance may lead the model to excessively copy the pre-trained model's intermediate representations, compromising flexibility. Therefore, it uses an attention map as a binary mask to guide which aspects of the intermediate features to distill and which to exempt, as shown in Eq. 21.

$$L_{KD} = \sum_{l=1}^{L} \| \mathbb{M}(\bar{f}_l) \odot (f_l - ReLU_m(\bar{f}_l)) \|_2, \tag{21}$$

where $\mathbb{M}(\cdot)$ is the module to calculate attention maps, $\odot$ is the Hadamard product, $ReLU_m$ is the function of margin ReLU activation.

AFC [82] also adopts a weighted distillation solution for intermediate feature maps to confine changes of important features in continual learning. It estimates the importance of each feature map by minimizing upper bound of loss increases and gives a theoretical demonstration with Taylor approximation.

Relation Alignment. TPCIL [76] introduces a topology-preserving loss to maintain the manifold's structure in feature space by penalizing alterations in class prototype relationships. It models the topological structure of the feature space using an Elastic Hebbian Graph (EHG) with nodes representing class prototypes and edges reflecting cosine distances between them. To retain prior knowledge, the newly formed EHG's topology is constrained to resemble previous one.

ERL [70] uses an Exemplar Relation Graph (ERG) to encode and preserve local relationships among cached exemplars, applying an exemplar relation distillation loss to maintain old knowledge. In ERG, each vertex represents an exemplar's feature in feature space, with edges being unit vectors connecting the vertices. The model defines relationships between feature space exemplars through angles formed by triplets of vertices within the ERG. By constraining angle alterations during continual learning, ERL regulates feature space changes, reinforcing the memory of previously learned information.

Co2L [83] employs self-distillation on instance-wise relations to sustain learned representations. It incorporates an asymmetric supervised contrastive loss for representation learning, aiming to keep relations intact within batch samples while learning new task. The method computes the similarity between each sample and other samples in the batch, creating

TABLE I: Summary of KD-integrated Continual Learning Methods

Paradigm	Methods	Distillation Level	Sub Distillation Type	KD Loss
KD as Regularization	LwF [45]	Logits	Probability Matching	Cross Entropy Loss
	BLD [75]			
	DMC [59]		Logits Matching	L2 Loss
	LwM [74]	Logits, Data	Implicit Data Alignment	L1 Loss
			Probability Matching	Cross Entropy Loss
	TPCIL [76]	Feature	Instance Feature Alignment	L2 Loss
			Relation Alignment	Topology-Preserving Loss
	PRD [77]		Relation Alignment	KL Divergence Loss
Combination of KD with Data Replay	iCaRL [46]	Logits	Probability Matching	Cross Entropy Loss
	EEIL [78]			
	BiC [93]			
	WA [95]			
	GD [80]			
	RDICL [94]			
	BI-R [114]			
	SS-IL [97]			
	Mnemonics [100]			
	COIL [61]			
	D+R [79]			
	FOSTER [104]			
	ILOS [66]			
	DualNet [120]			KL Divergence Loss
	DyTox [103]			
	LVT [98]			
	RPS-Net [102]			
	DCR [121]			
	DER++ [86]		Logits Matching	L2 Loss
	XDER [60]			
	GCR [101]			
	CF-IL [92]			
	LUCIR [96]	Feature	Instance Feature Alignment	Cosine Similarity Loss
	GeoDL [84]			
	L2I [122]			L2 Loss
	MDMT-R [118]			
	ABD [90]			
	PODNet [81]		Instance Feature Alignment	L2 Loss
			Hidden Feature Alignment	POD-Spatial Loss
	TwF [119]		Hidden Feature Alignment	L2 Loss
	AFC [82]			
	Co2L [83]		Relation Alignment	Instance-wise Relation Distillation loss
	ERL [70]			Local Relation Loss
	R-DFCIL [91]	Logits, Feature	Logits Matching	L1 Loss
			Relation Alignment	Local Relation Loss
	MBP [85]		Probability Matching	Label Priority-Preserving loss
			Relation Alignment	Instance Neighborhood-Preserving Loss
	SNCL [123]		Logits Matching	L2 Loss
			Hidden Feature Alignment	
	OCD-Net [87]		Logits Matching	L2 Loss
			Relation Alignment	Contrastive Relation Distillation loss
	MeRGAN [88]	Data	Explicit Data Alignment	L2 Loss
	PGMA [89]	Logits, Data	Logits Matching	L2 Loss
			Explicit Data Alignment	
			Implicit Data Alignment	
Combination of KD with Feature Replay	GFR [105]	Feature	Instance Feature Alignment	L2 Loss
	IL2A [107]			
	PASS [106]			
	FRoST [108]			
	Fusion [112]			
	NCDwF [124]			
	SSRE [62]			
	PRAKA [111]			
	FeTrIL [110]			
	MEIL [113]	Logits, Feature	Probability Matching	Cross Entropy Loss
			Instance Feature Alignment	Cosine Similarity Loss

a similarity vector. This similarity relationship between the new and old model serves to distill knowledge from previous tasks. As this process is grounded in learned features, it is considered a feature-level distillation method.

MBP [85] focuses on preserving the model's feature extraction behavior by maintaining the relative ordering of distances between instances in the feature space with an instance neighborhood-preserving loss. MBP calculates pairwise distances among all instances, identifying the top K nearest neighbors for each. By ensuring the order of proximal points remains consistent, MBP upholds semantic instance relationships while permitting development of more adaptable features, enhancing the model's plasticity.

OCD-Net [87] integrates internal instance structure information from the teacher model using supervised contrastive learning [125]. It employs a contrastive relation distillation loss to cluster embeddings of the same class and separate those of different classes, utilizing cosine distance with temperature scaling. The teacher's embedding serves as an anchor, guiding the student's embedding toward the same-class embeddings found in the teacher model.

R-DFCIL [91] employs an angle-wise relational KD loss to learn representations for new classes while preserving representations for old classes. It achieves this by stabilizing the mutual spatial relationships of a triplet of new data within the feature space, which is transformed using a learnable linear layer. The relation KD to a triplet (x_a, x_b, x_c) is as follows:

$$L_{KD} = \|\cos \angle t_a t_b t_c - \angle s_a s_b s_c\|_1, \tag{22}$$

$$\cos \angle r_a r_b r_c = \langle e^{ab}, e^{cb} \rangle, e^{ij} = \frac{r_i - r_j}{\|r_i - r_j\|_2}, \tag{23}$$

where t is transformed representations from old model, s is transformed representations from new model.

C. Data-level Distillation

Data-level distillation can be split into two types: Explicit Data Alignment and Implicit Data Alignment. Explicit Data Alignment entails distilling synthetic data produced by generative models. Contrastingly, Implicit Data Alignment focuses on distilling underlying information within the data, like attention maps or latent codes from generative models.

Explicit Data Alignment. MeRGAN [88] ensures memory retention in GANs with a L2 loss during continual learning by aligning replay data generated by the model. Specifically, when given the same category and latent information as input to the generative model during learning new tasks, the content of the data generated by the new generative model should be consistent with that generated by the old generative model. This type of data alignment is a form of data distillation and has been adopted by other generative-model-based continual learning approaches, such as [89], [126].

Implicit Data Alignment. LwM [74] confronts forgetting by stabilizing the model's attention. Using Grad-CAM [127], it calculates attention maps and preserves attention regions for old tasks while learning new ones, ensuring feature extracting capabilities are retained for prior knowledge. Additionally, LwM implements logits-level distillation similar to LwF, reinforcing old task knowledge consolidation. It follows Equation 24 to distill attention map where $\hat{Q}$ and Q are vectorized attention map from old and new learning model:

$$L_{KD} = \|\frac{\hat{Q}}{\|\hat{Q}\|_2} - \frac{Q}{\|Q\|_2}\|_1. \tag{24}$$

PGMA [89] alleviates forgetting by leveraging a VAE-based generator [128] to synthesize previous task data. The VAE features an encoder that compresses inputs into a latent representation and a decoder that reconstructs the inputs from this code. Since the generator itself must also continuously learn to synthesize new task data, PGMA applies distillation on the encoder's sample-specific latent code with a L2 loss during VAE's continual learning. By regulating the latent space, this strategy helps curb forgetting within the VAE model.

D. Discussion

For these three levels KD, logits-level distillation constrains the final output from the classification head, providing rich information, but its effectiveness depends on the quality of the teacher model and the form of knowledge transfer. Feature-level distillation focuses on the feature space, allowing distillation of different dimensions features and spatial relationships, which benefits semantic retention but lacks memory retention for the classification head. Data-level distillation is less common and primarily used for generating synthetic replay data or some special scenarios.

V. Experiments

In this section, we experiment with ten KD-integrated continual learning methods on three image classification datasets and explore the role of KD in continual learning with extensive ablation studies.

A. Datasets

We select three image classification datasets widely used in continual learning field: CIFAR-100, TinyImageNet, and ImageNet-100, encompassing a range of image resolutions from 32×32, 64×64, to 224×224 pixels respectively. CIFAR-100 [129], drawn from the "80 Million Tiny Images" collection [130], comprises 100 different categories, each with 500 training images and 100 testing images at a resolution of 32×32 pixels. ImageNet [131] is a large-scale dataset with 1.28 million training images and 50,000 validation images, spread across 1,000 categories. The ImageNet dataset has spawned various derivatives, which are sometimes inconsistently named in continual learning studies. We provide a unified nomenclature for clarity: ImageNet-1000, or ImageNet-Full, refers to the complete dataset, whereas ImageNet-100, introduced by iCaRL [46] and also known as ImageNet-Subset, features a fixed random seed shuffle of the first 100 ImageNet categories. Studies [82], [98], [103], [100], [81] commonly consider ImageNet-100 as a standard benchmark following the iCaRL setting. TinyImageNet [132], a downsized derivative of ImageNet created by Stanford University, has 200 categories with images downsampled to 64×64 pixels.

TABLE II: Comparison on Different Datasets. A growing memory with 20 exemplars each class is adopted for replay-based methods.

Methods	CIFAR-100		TinyImageNet		ImageNet-100	
	(10/10)	11/50-5	(10/20)	11/100-10	(10/10)	(11/50-5)
LwF	25.40±1.26	18.54±2.42	19.63±1.21	13.34±1.16	26.08±1.19	15.10±0.77
LwM	25.84±1.62	18.11±1.45	20.29±2.02	11.96±1.67	26.50±0.66	14.88±0.33
iCaRL	43.29±1.88	54.45±0.91	36.33±0.61	44.86±0.62	48.12±1.50	57.94±0.93
EEIL	39.86±1.31	44.21±1.50	29.23±0.80	33.32±1.76	36.72±1.56	37.12±1.15
BiC	**48.24±2.22**	52.20±3.11	**39.52±1.83**	40.50±0.79	**52.78±1.10**	50.73±1.65
LUCIR	45.44±1.77	46.66±1.77	31.65±1.02	35.06±0.87	41.48±1.21	46.60±1.28
SS-IL	44.08±3.16	50.18±0.59	37.23±0.79	38.39±0.23	50.33±0.75	45.11±1.37
IL2A	31.79±3.78	53.46±0.75	32.61±1.72	38.25±0.97	30.73±1.48	56.79±0.79
PASS	35.34±0.64	53.69±0.29	31.17±0.57	44.09±0.49	34.17±0.39	61.39±0.54
PRAKA	43.16±0.73	**59.45±0.48**	36.02±0.62	**49.02±0.79**	35.56±0.98	**61.80±0.78**

B. Protocols and Scenarios

Our experiments focus on the CIL, using an offline and balanced protocol to evaluate different baseline methods. In balanced and offline CIL setup, two main strategies simulate data increment scenarios: the first divides the dataset into tasks with an equal number of classes for sequential learning. The second starts with preliminary base training on a subset of classes, followed by increments using remaining classes. To clearly describe these scenarios, we adopt the notation from [8]. Specifically, the first scenario is notated as (A/B), where 'A' indicates the number of tasks and 'B' signifies the number of classes per task. For example, splitting CIFAR-100 into 10 sequential tasks with 10 classes each is notated as (10/10). The second scenario, involving base training followed by class increments, is expressed as (A/C-B). Here, 'A' stands for the total number of tasks, 'C' represents the initial base training class count (the first task), and 'B' indicates the number of classes for each subsequent incremental task. For instance, in the CIFAR-100 dataset, (11/50-5) denotes partitioning into 11 tasks—with initial base training on 50 classes, followed by an even distribution of the remaining 50 classes into 10 incremental tasks, each with 5 classes.

C. Metrics

There are many evaluation metrics used in continual learning for image classification tasks. iCaRL [46] introduces *Average Incremental Accuracy* (AIA), which captures the average of aggregate average accuracy after completing the learning process across all tasks. Meanwhile, GEM [133] introduces *Average Accuracy* (AA), *Backward Transfer* (BWT) and *Forward Transfer* (FWT) to assess the extent of catastrophic forgetting and the transferability of knowledge. Furthermore, RWalk [134] proposes *Forgetting Measure* (FM) and *Intransigence Measure* (IM) to evaluate the model's average forgetting and inability to learn new tasks.

We perform a statistical analysis of the papers listed in Table I and found that *Average Accuracy* is the most prevalently utilized metric for Class-Incremental Learning (CIL). Other metrics, which are originally designed for Task-Incremental Learning (TIL), demonstrate limited applicability to CIL [8]. Consequently, we adopt *Average Accuracy* as the main metric. *Average Accuracy* evaluates average performance after learning t-th task which can be defined as

$$AA_t = \sum_{i=1}^{t} \frac{C_i}{C_{1:t}} a_{t,i} \tag{25}$$

where $a_{t,i}$ is the accuracy of task i evaluated on the test set after learning task t and $i \leq t$. C_i denotes the number of classes contained in the task i , while $C_{1:t}$ represents the cumulative sum of classes encompassed by all tasks learned up to the task t.

D. Baselines

We have chosen ten KD-integrated continual learning methods from Table I to facilitate an exhaustive comparative analysis. These selected methods are LwF [45], LwM [74], IL2A [107], PASS [106], PRAKA [111], iCaRL [46], EEIL [78], BiC [93], LUCIR [96], and SS-IL [97]. Each method encapsulates distinct facets of the KD-integrated continual learning discussed in Sec. III and IV. More specifically, LwF and LwM fall under the "KD as Regularization" paradigm. iCaRL, EEIL, BiC, LUCIR, and SS-IL belong to the "Combination of KD with Data Replay" paradigm, while PASS, IL2A, and PRAKA are part of the "Combination of KD with Feature Replay" paradigm. Additionally, when categorized by the type of knowledge source, LwF, iCaRL, EEIL, BiC and SS-IL are grouped as Logits-Level distillation techniques. LUCIR, PASS, and IL2A are categorized under Feature-Level distillation, while LwM is considered as a Data-Level distillation method.

E. Training Details

We adopt ResNet-18 [135] as the standard backbone network across all baseline methods due to its widespread use in image classification and continual learning benchmarks. ResNet-18, tailored for large-scale and high-resolution image datasets such as ImageNet, is modified to better suit smaller resolution datasets in our study. We adjust the initial convolutional layer by reducing the kernel size from 7×7 to 3×3 and omitting the initial max-pooling operation. The adapted ResNet-18 is employed for training on CIFAR-100 and TinyImageNet, while the original architecture remains for ImageNet-100 to accommodate its high-resolution images.

TABLE III: Comparison on KD effect for 10/10, 11/50-5 incremental settings, exemplars-based methods use a growing memory with 20 exemplars each class for all scenarios.

Methods	(10/10)		(11/50-5)	
	with KD	w/o KD	with KD	w/o KD
LwF	**24.99±1.12**	11.95±0.93	**18.17±2.40**	7.02±0.73
LwM	**26.10±1.48**	24.99±1.12	**18.60±1.19**	18.19±2.40
iCaRL	44.53±2.36	**48.08±1.24**	**53.74±0.90**	50.79±0.37
EEIL	**39.92±1.76**	39.54±1.70	**42.70±0.82**	42.63±1.43
BiC	**48.09±2.70**	45.12±2.19	**55.21±0.82**	47.82±0.78
LUCIR	44.50±1.12	**44.87±0.70**	**47.35±1.78**	46.39±1.71
SS-IL	**44.57±1.80**	41.02±0.53	**49.80±0.43**	42.87±0.75
IL2A	**33.50±2.59**	9.06±0.23	**51.39±1.27**	4.62±0.13
PASS	**35.10±0.82**	8.92±0.33	**53.45±0.86**	4.62±0.16
PRAKA	**43.57±0.59**	9.01±0.35	**59.27±0.42**	4.62±0.14

We implement all the methods selected in Sec. V-D using the FACIL [8] framework, which is implemented in PyTorch. For all methods except IL2A, PASS and PRAKA, we use an SGD optimizer with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 0.0002. IL2A, PASS and PRAKA employ an Adam optimizer with an initial learning rate of 0.001 and a momentum of 0.9. The learning rate is decreased every 45 epochs by a factor of 0.1 for IL2A, PASS and PRAKA. We use the herding sampling strategy for all exemplars-based methods. All experimental results are obtained by conducting five trials and calculating the mean and standard deviation of the results.

F. Results

In this section, we begin by conducting experimental comparisons of baseline methods within different datasets. Subsequently, we delve into detailed experiments and in-depth analysis of the role that KD plays in mitigating forgetting.

1) On Datasets: In this section, we assess baseline methods on datasets with varying resolutions: CIFAR-100 (100 classes, 32×32), TinyImageNet (200 classes, 64×64), and ImageNet-100 (100 classes, 224×224). We compare performances in two scenarios: a 10-task scenario without base training and an 11-task scenario with base training. For data replay based methods, we employ a growing memory that holds 20 exemplars per task.

Table II presents the empirical outcomes of several baseline strategies across datasets with varying resolutions: CIFAR-100, TinyImageNet, and ImageNet-100. In the 10-task scenario without base training, the BiC approach outperforms its counterparts on all datasets. For the 11-task scenario with base training, PRAKA stands out as the top performer across all datasets. In scenarios with or without base training, methods in "Combination KD with Data replay" paradigm achieve good performance. In the scenario without base training, methods in "Combination of KD with Feature replay" paradigm show slightly inferior performance compared to those employing the data replay paradigm. However, in the context of base training, the performance of feature replay methods improves significantly, with PRAKA outperforming methods using the data replay paradigm across all datasets. In contrast, methods in "KD as Regularization" paradigm, which solely employ KD techniques, perform considerably worse than the other two paradigms on all datasets. The performance of LwF and LwM in scenarios with base training is lower than in scenarios without base training. This discrepancy is attributed to the local classification loss used by LwF and LwM during the acquisition of new knowledge, where only the classification head of the new task is learned. Conversely, other methods generally achieve better results with the inclusion of base training, except for BiC and SS-IL on ImageNet-100, where they perform better without it. Notably, the relative benefit of base training for BiC and SS-IL diminishes as the resolution of the dataset increases.

TABLE IV: Comparison on different KD losses. CE is Cross Entropy loss, KL represents KL divergence loss, L2 is L2 distance loss, CS represents Cosine Similarity loss.

Methods		10/10		11/50-5	
Finetuning		11.82±0.36		7.22±0.63	
Logits-level	+ CE	22.99±0.60	(+11.17)	13.22±1.39	(+6.00)
	+ KL	30.58±1.17	(+18.76)	22.26±1.48	(+15.04)
	+ L2	**33.51±1.41**	(+21.69)	**28.82±2.71**	(+21.60)
Feature-level	+ L2	12.18±0.72	(+0.36)	8.44±0.66	(+1.22)
	+ CS	**15.55±1.46**	(+3.73)	**15.80±1.09**	(+8.58)

2) On KD Effect: This section delves into KD's role in continual learning methods by conducting ablation studies that omit the KD loss. We analyze ten approaches, predominantly using a single distillation loss function, except for LwM, which employs two forms of distillation loss. We remove the logits-level distillation loss used by LwF, iCaRL, and BiC to assess its impact. For EEIL, we eliminate its task-wise KD loss employed in both balanced and unbalanced training phases. SS-IL is similarly adjusted by removing its task-wise KD loss that complements separated softmax. For IL2A, PASS and PRAKA, we abrogate the L2-based feature distillation loss crucial for instance feature alignment. LUCIR's modification entails removing the cosine feature distillation loss. In LwM, we exclude only the attention map distillation loss—which is considered as implicit data distillation—to isolate the effect of the attention distillation, keeping its logits-level distillation intact.

Table III highlights how KD impacts different continual learning approaches on CIFAR-100 in (10/10) and (11/50-5) scenarios. From the table, it is evident that KD plays a crucial role in methods following the "Combination of KD with Feature Replay" paradigm. Whether in scenarios with or without base training, the removal of KD leads to a sharp decline in performance. Similarly, KD also plays a significant role in the LwF method, while in the LwM method, the use of attention map distillation can slightly enhance the resistance to forgetting. The results for methods under the "Combination of KD with Data Replay" paradigm, however, appear somewhat different. The table shows that in scenarios with base training, KD significantly contributes to mitigating forgetting, with performance drops observed across all methods once distillation is removed. In scenarios without base training, EEIL, BiC, and SS-IL also exhibit performance declines upon the removal of KD. Conversely, iCaRL and LUCIR show performance improvements, with iCaRL displaying a significant increase and LUCIR showing a smaller increase. Among all methods under the "Combination of KD with Data Replay" paradigm, BiC and SS-IL exhibit a strong dependence on distillation,

TABLE V: Comparison on different KD losses with data replay. CE is Cross Entropy loss, KL represents KL divergence loss, L2 is L2 distance loss, CS represents Cosine Similarity loss.

(a) Global Classification

Methods		(10/10)		(11/50-5)	
Replay		41.10±1.18		42.25±1.50	
Logits-level	+ CE	**38.41±0.78**	(-2.69)	40.09±1.03	(-2.16)
	+ KL	33.99±0.90	(-7.11)	40.76±1.03	(-1.49)
	+ L2	34.35±0.97	(-6.75)	**41.66±1.13**	(-0.59)
Feature-level	+ L2	**42.21±0.81**	(+1.11)	43.52±0.92	(+1.27)
	+ CS	41.39±1.69	(+0.29)	**44.29±1.57**	(+2.04)

(b) Separated Softmax Classification

Methods		10/10		11/50-5	
Replay		45.09±0.93		48.05±0.76	
Logits-level	+ CE	44.85±1.55	(-0.24)	49.74±1.78	(+1.69)
	+ KL	45.34±1.42	(+0.25)	52.33±0.78	(+4.28)
	+ L2	**46.36±0.84**	(+1.27)	**55.56±0.46**	(+7.51)
Feature-level	+ L2	45.25±1.29	(+0.16)	48.55±0.80	(+0.50)
	+ CS	**45.44±1.41**	(+0.35)	**49.25±1.03**	(+1.20)

with noticeable performance declines in both scenarios once distillation is deactivated.

3) On KD Losses: To evaluate the effectiveness of different KD losses in mitigating forgetting, we conducted standalone assessments of various KD losses without using any additional techniques for preventing forgetting. We evaluate cross-entropy, KL divergence, L2 distance loss at the logits level, as well as L2 distance-based and cosine similarity-based instance feature alignment losses. For the classification loss used in continual learning, we adopted an approach followed LwF [45], where only the classification head for the current task is trained, while the heads of previous tasks are solely involved in distillation, as using a global classification loss without data from previous tasks would cause severe classification bias issues and significantly reduce the effectiveness of KD.

Table IV presents the ability of different KD losses to mitigate forgetting. It can be observed that all KD losses exhibit some degree of forgetting mitigation. Among them, the logits-level KD losses demonstrate significantly better forgetting mitigation compared to feature-level KD losses. This is because logits-level KD losses impose constraints on the final classification head, whereas feature-level KD losses only add constraints to the feature extraction part of the network. Consequently, logits-level KD losses are more effective than feature-level KD losses. This finding indicates that in continual learning for image classification tasks, adding forgetting mitigation constraints to the classification head can further enhance the overall model's ability to mitigate forgetting. Among all logits-level KD losses, the L2 distance loss, which has a stronger constraint capability, performs better in mitigating forgetting compared to KL divergence, with cross-entropy distillation loss being the least effective. For feature-level KD losses, the cosine similarity loss, which contains more semantic information, is superior to the L2 distance loss in mitigating forgetting.

4) On KD with Data Replay: To further understand KD's role when combined with data replay and explore the effects of different KD losses, we compare several KD losses alongside a basic replay paradigm using herding algorithm for a growing memory with 20 exemplars per class. We evaluate the same KD losses as Sec. V-F3. For the classification loss in continual learning, we employed the global classification loss commonly used in most replay methods [46], [78], [96]. Through this comparison, we aim to gain deeper insight into each KD loss's specific contribution to continual learning with data replay.

Table Va shows that integrating KD with data replay. Surprisingly, we find that regardless of whether base training is applied, incorporating logits-level KD consistently leads to performance degradation, aligning with observations from other research efforts [136], [8], [137]. This negative impact is particularly pronounced in the absence of base training, where logits-level KD causes a significant decline in performance. Without base training, the KL divergence KD loss causes the greatest harm, while the cross-entropy KD loss shows slightly lesser damage. In contrast, feature-level KD demonstrates some improvement in mitigating forgetting across both scenarios, though the extent of this improvement is limited. The positive effect is slightly more pronounced with base training, and the cosine similarity loss appears advantageous for retaining learned features. However, without base training, the effect of cosine similarity loss to maintain memory is inferior to that of L2 loss.

Motivated by SS-IL [97] and Sec. V-F3, we hypothesize that this could also be due to classification bias introduced by the classification head. To verify our hypothesis, we follow SS-IL and use a separated softmax loss to learn classification head, namely using replay data to jointly train all old task classification heads, while new task data is used exclusively to train the new task classification head. Table Vb confirms our hypothesis that the classification bias introduced by the classification head does indeed affect the performance of KD. Surprisingly, even without using KD, data replay with separated softmax outperforms data replay with global classification. When KD is added, all configurations, except for using cross-entropy in the (10/10) scenario, resulted in a positive impact on preventing forgetting. The improvement is particularly notable under base training scenario, with logits-level KD showing significant performance gains. This strongly suggests that classification bias affects KD effectiveness, especially for logits-level KD. Within logits-level KD, L2 loss remains the most effective, followed by KL divergence, with cross-entropy being the least effective. Both L2 loss and KL divergence even outperform the task-wise distillation used in SS-IL. Furthermore, logits-level KD is consistently more effective than feature-level KD, which is consistent with the results obtained in Sec. V-F3. Additionally, both BiC and SS-IL, which tackle the classification bias problem, are the most reliant on KD within the "Combination of KD with Data Replay" paradigm. This further supports the conclusion that classification bias significantly impacts the effectiveness of KD.

VI. KD In Other Tasks

Beyond image classification, KD has been extensively utilized in continual learning for other vision tasks such as object detection and semantic segmentation. To discuss the similarities and differences compared to KD application in

TABLE VI: Comparison of Methods for Continual Object Detection with KD.

Methods	Feature Distill.	RCN Distill.	RPN Distill.	VOC (2/15-5) 1-15	VOC (2/15-5) 16-20	VOC (2/15-5) 1-20	VOC (6/15-1) 1-15	VOC (6/15-1) 16-20	VOC (6/15-1) 1-20
Faster ILOD	✓	✓	✓	**70.10**	60.13	**67.61**	**67.21**	**45.18**	**61.70**
	✓		✓	1.90	54.63	15.08	3.42	11.29	5.39
	✓	✓		69.57	**61.50**	67.55	65.20	44.93	60.13
		✓	✓	69.32	59.90	66.97	66.64	44.90	61.20
				9.54	54.08	20.68	1.64	9.34	3.49
ABR	✓	✓	✓	**71.07**	60.77	**68.49**	**69.09**	**48.64**	**63.98**
	✓		✓	40.10	56.21	44.13	30.30	27.65	29.64
	✓	✓		70.24	**62.36**	68.27	68.05	48.57	63.18
		✓	✓	69.94	61.68	67.88	68.26	49.09	63.46
				9.54	54.08	20.68	1.64	9.34	3.49

image classification and evaluate the effectiveness of KD in mitigating forgetting across these different tasks, we conduct comparison experiments to assess impact of KD in these tasks.

A. Methods and Experimental Settings

We select four methods incorporating KD: two for continual object detection [138], [139] and two for continual semantic segmentation [140], [141]. For continual object detection tasks, Faster ILOD [138] utilizes three distillation losses: one applied to the feature maps using normalized adaptive distillation with an L1 loss, another applied to the Region Proposal Network (RPN) outputs using an L2 loss, and a third applied to the outputs of classification and box prediction head (RCN) using an L2 loss. Building upon of Faster ILOD, ABR [139] introduces an attentive RoI distillation loss to enhance feature distillation and an inclusive distillation loss to improve the distillation of RCN outputs. For continual semantic segmentation tasks, REMINDER [140] applies a class similarity KD loss on the segmentation output to revise knowledge for old classes that are likely to be forgotten due to their similarity to new classes, combined with local pooled outputs distillation (Local POD) [142] applied to features in intermediate layers. LGKD [141] also applies a Local POD loss to the intermediate layer features of the network. For the segmentation outputs, it introduces a label-guided KD loss to help the model correctly establish the correspondence between new classes and the background during distillation.

We conduct ablation studies on two continual object detection methods, evaluating the distillation of features, RPN, and RCN outputs individually. For continual semantic segmentation, we assess feature and segmentation output distillation. These experiments aim to measure the effectiveness of each distillation loss in mitigating forgetting. To isolate the effect of KD, we removed the replay module from the ABR method. Using the Pascal VOC dataset, we test two continual learning scenarios for both tasks: (2/15-5) and (6/15-1). Performance is measured using mean Average Precision (mAP) for continual object detection and mean Intersection over Union (mIoU) for continual semantic segmentation. For both tasks, we calculate the average metrics for the initial Task 1-15 categories, subsequent incremental tasks, and the final average metrics after learning all categories. All experiments are conducted based on the original code of the respective methods.

B. Results and Discussion

It can be observed that for both continual object detection and continual semantic segmentation, the distillation losses can be broadly categorized into feature-level distillation and output-level distillation (similar to image classification tasks where it is termed logits-level). In continual object detection, Feature Distillation and RPN Distillation can be considered as feature-level distillation, while RCN Distillation is a distillation of the model's output. To distinguish it from logits-level distillation in image classification tasks, we refer to it as output-level distillation. Similarly, for the two continual semantic segmentation methods, the Local POD loss used falls under feature-level KD, while the distillation of the final segmentation results falls under output-level distillation.

Table VI shows the effects of using different distillation methods in Faster ILOD and ABR, while Table VII presents the effects of different distillation losses in REMINDER and LGKD. Both sets of results demonstrate the crucial role of KD in mitigating forgetting across both tasks. For continual object detection tasks, it can be observed from Table VI that the RCN distillation loss, which distills the final outputs, plays a more important role in mitigating forgetting compared to the Feature Distillation and RPN Distillation, both belonging to feature-level distillation and show a smaller impact. For continual semantic segmentation tasks, both feature-level and output-level KD can alleviate forgetting to some extent. Particularly in the more complex continual learning scenario (6/15-1), feature-level KD significantly outperforms output-level KD in terms of mitigating forgetting.

VII. Trends of Continual Learning with KD

This section mainly highlights future trends of continual learning with KD from three distinct views.

A. KD with High-Quality Knowledge

KD has shown promise in mitigating catastrophic forgetting in continual learning, but there is significant potential for further refinement. Effective knowledge transfer hinges on the quality of the knowledge being distilled. High-quality knowledge transfer is crucial for enhancing the performance of knowledge distillation in continual learning.

Recent advancements highlight various methods that aim to improve the quality of distilled knowledge. DER++ [86] matches logits sampled throughout the optimization trajectory. XDER [60] implant logits to prevent teacher model

TABLE VII: Comparison of Methods for Continual Semantic Segmentation with KD.

Methods	Seg. Output Distill.	Feature Distill.	VOC (2/15-5)			VOC (6/15-1)		
			1-15	16-20	1-20	1-15	16-20	1-20
REMINDER	✓	✓	**74.23**	**46.69**	**68.41**	**64.89**	22.12	**55.63**
	✓		67.21	40.28	61.86	22.14	5.24	21.09
		✓	73.49	48.53	68.32	62.39	**23.82**	54.12
			65.48	38.43	60.19	28.63	5.73	25.85
LGKD	✓	✓	**77.98**	55.26	**73.20**	**69.95**	**28.29**	**60.94**
	✓		76.95	**55.76**	72.59	23.31	7.91	22.48
		✓	75.02	49.07	69.55	61.73	21.64	51.82
			68.35	40.39	62.72	30.36	4.75	26.87

from losing valuable secondary information. OCD-Net [87] learns knowledge from an online teacher trained by random momentum update. These methods focus on retaining the critical historical knowledge to improve memory retention. In the field of KD, numerous methods aim to achieve more efficient knowledge transfer, such as conveying the knowledge embedded in logits across multi levels [143], distinct influences of target class and non-target class knowledge in the logits [144], respecting transfer gap with inverse probability weighting distillation [145]. These diverse approaches highlight the potential of various distillation methods to enhance the quality of knowledge transfer. Future research about continual learning with KD should prioritize the development of more sophisticated techniques for discerning high-quality knowledge.

B. KD Tailored to Specific Tasks

Continual learning has expanded from its initial focus on classification tasks to encompass a wide range of other tasks, such as object detection [146], [138], [147], [139], [148], semantic segmentation [142], [140], [141] in vision domain, language learning [149], [150], machine translation [151], [152], intent detection [153], [154] and named entity recognition [155] in natural language processing (NLP) domain.

KD also plays a crucial role in mitigating forgetting across these various tasks. Methods in these continual learning domains employ KD in ways similar to image classification, utilizing distillation at the feature level or task output level (logits level in image classification task). However, due to the specificity of problems within the task domains, many distillation methods are specifically designed based on the characteristics of the tasks. For example, [147] proposes RCN distillation for the output of classification and bounding box regression network during continual object detection. [138] mitigate forgetting by applying KD to the features within the region proposal network (RPN) of the object detection framework. ABR [139] uses an Attentive RoI (Region-of-Interesting) Distillation, which uses spatial attention from RoI features. LGKD [141] proposes Label Guided Knowledge Distillation to assists the model in correctly establishing the correspondence between new classes and the background during continual semantic segmentation. In NLP domain, DnR [149] performs three different distillation methods to match the internal representations during continual language learning. CL-NMT [151] proposes a dynamic KD loss to balance the attention between old knowledge and new knowledge during continual learning for neural machine translation. In intent detection task, CID [153] employs Hierarchical Knowledge Distillation to preserve the feature and probability distributions of previous classes. In the future, KD will continue to play a significant role in these continual learning tasks. Designing distillation losses that are more tailored to the characteristics of these tasks will be essential. Additionally, identifying the key components of tasks that are prone to forget, and leveraging KD to mitigate forgetting, will be crucial.

C. KD with Better Teachers

In recent years, continual learning based on pre-trained model (PTM) and large language model (LLM) has garnered increasing attention. KD is a naturally advantageous method for mitigating forgetting for both PTM-based and LLM continual learning. This is because KD follows the Teacher-Student framework, where PTM and LLM already possess rich knowledge, effectively starting their continual learning from a highly experienced teacher.

However, current PTM-based continual learning methods [156], [157], [158], [159] primarily integrate Parameter-Efficient Fine-Tuning (PEFT) techniques [160], [161], [162], [163], achieving continual learning for new tasks by fine-tuning a small number of task-related parameters while keeping the foundational knowledge of the PTM unchanged. There is limited exploration of KD in continual learning based on PTM. Future research could explore how to utilize KD to maintain the foundational knowledge of PTM while minimizing the parameter amount changes introduced by PEFT during new knowledge acquisition.

Regarding LLM, KD has already played a significant role in LLM training by bridging the performance gap between proprietary commercial LLM and open-source LLM due to differences in the volume of pre-training data [164]. For LLM continual learning, KD also serves as an important strategy for mitigating forgetting throughout the continual learning process, encompassing Continual Pre-Training, Domain-Adaptive Pre-Training, and Continual Fine-Tuning [165]. In the future, continual learning for LLM will not only involve continually learn unknown tasks with new data but also aim to achieve overall knowledge growth by distilling knowledge from expert LLMs with different architectures into a unified LLM. Therefore, heterogeneous architectures KD [166] will become particularly important. Furthermore, multi-modal LLMs are anticipated to be the trend, making cross-modal KD [167], [168] a key technology for the continual learning of these multi-modal models.

VIII. Conclusion

Our study performs an in-depth investigation of continual learning methods utilizing KD. We provide a detailed analysis of how KD is utilized in continual learning methods, categorizing its application into three distinct paradigms. Besides, we classify these methods based on knowledge sources and present an overview of different distillation losses to prevent forgetting. Extensive experiments on ten KD-integrated methods across three datasets in varied scenarios highlight KD's role in continual learning, and we also substantiate that classification bias can impair the performance of KD and separated softmax loss can alleviate this issue. Additionally, we explore the effectiveness of knowledge distillation to mitigate forgetting in other vision tasks and discuss the future directions for continual learning with KD. This paper aims to deepen the understanding of KD's impact in continual learning and to inform ongoing research in the domain.

References

[1] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[2] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 12–25, 2015.

[3] M. Sayed-Mouchaweh and E. Lughofer, *Learning in non-stationary environments: methods and applications*. Springer Science & Business Media, 2012.

[4] Z. Chen and B. Liu, *Lifelong machine learning*, vol. 1. Springer, 2018.

[5] R. Wang, Y. Bao, B. Zhang, J. Liu, W. Zhu, and G. Guo, "Anti-retroactive interference for lifelong learning," in *Eur. Conf. Comput. Vis.*, pp. 163–178, Springer, 2022.

[6] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *arXiv preprint arXiv:1812.00420*, 2018.

[7] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[8] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. Van De Weijer, "Class-incremental learning: survey and performance evaluation on image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5513–5533, 2022.

[9] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Deep class-incremental learning: A survey," *arXiv preprint arXiv:2302.03648*, 2023.

[10] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, Elsevier, 1989.

[11] R. Aljundi, M. Rohrbach, and T. Tuytelaars, "Selfless sequential learning," in *Int. Conf. Learn. Represent.*, 2019.

[12] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[13] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in Psychology*, vol. 4, p. 504, 2013.

[14] S. T. Grossberg, *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control*, vol. 70. Springer Science & Business Media, 2012.

[15] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *arXiv preprint arXiv:2302.00487*, 2023.

[16] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, 2021.

[17] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, 2021.

[18] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Int. Conf. Comput. Vis.*, pp. 3713–3722, 2019.

[19] L. Zhang, C. Bao, and K. Ma, "Self-distillation: Towards efficient and compact neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4388–4403, 2021.

[20] H. Qu, H. Rahmani, L. Xu, B. Williams, and J. Liu, "Recent advances of continual learning in computer vision: An overview," *arXiv preprint arXiv:2109.11369*, 2021.

[21] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, 2021.

[22] E. Belouadah, A. Popescu, and I. Kanellos, "A comprehensive study of class incremental learning algorithms for visual tasks," *Neural Networks*, vol. 135, pp. 38–54, 2021.

[23] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[24] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[25] T. L. Hayes, G. P. Krishnan, M. Bazhenov, H. T. Siegelmann, T. J. Sejnowski, and C. Kanan, "Replay in deep learning: Current approaches and missing biological elements," *Neural Computation*, vol. 33, no. 11, pp. 2908–2950, 2021.

[26] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information Fusion*, vol. 58, pp. 52–68, 2020.

[27] Z. Ke and B. Liu, "Continual learning of natural language processing tasks: A survey," *arXiv preprint arXiv:2211.12701*, 2022.

[28] M. Biesialska, K. Biesialska, and M. R. Costa-jussà, "Continual lifelong learning in natural language processing: A survey," in *COLING*, pp. 6523–6541, 2020.

[29] P. Zhang and S. Kim, "A survey on incremental update for neural recommender systems," *arXiv preprint arXiv:2303.02851*, 2023.

[30] Y.-C. Hsu, J. Smith, Y. Shen, Z. Kira, and H. Jin, "A closer look at knowledge distillation with features, logits, and gradients," *arXiv preprint arXiv:2203.10163*, 2022.

[31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[32] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[33] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 126–136, 2018.

[34] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *Int. Conf. Learn. Represent.*, 2018.

[35] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, "Unifying distillation and privileged information," in *Int. Conf. Learn. Represent.*, 2016.

[36] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Int. Conf. Comput. Vis.*, pp. 1365–1374, 2019.

[37] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4320–4328, 2018.

[38] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, "Online knowledge distillation with diverse peers," in *AAAI Conf. Artif. Intell.*, vol. 34, pp. 3430–3437, 2020.

[39] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *AAAI Conf. Artif. Intell.*, vol. 34, pp. 5191–5198, 2020.

[40] W. Son, J. Na, J. Choi, and W. Hwang, "Densely guided knowledge distillation using multiple teacher assistants," in *Int. Conf. Comput. Vis.*, pp. 9395–9404, 2021.

[41] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisit knowledge distillation: a teacher-free framework," *arXiv preprint arXiv:1909.11723*, 2019.

[42] J. Wang, W. Bao, L. Sun, X. Zhu, B. Cao, and S. Y. Philip, "Private model compression via knowledge distillation," in *AAAI Conf. Artif. Intell.*, vol. 33, pp. 1190–1197, 2019.

[43] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *IEEE Symposium on Security and Privacy*, pp. 582–597, IEEE, 2016.

[44] H. Wang, Y. Deng, S. Yoo, H. Ling, and Y. Lin, "Agkd-bml: Defense against adversarial attack by attention guided knowledge distillation and bi-directional metric learning," in *Int. Conf. Comput. Vis.*, pp. 7658–7667, 2021.

[45] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, 2017.

[46] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 2001–2010, 2017.

[47] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[48] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Int. Conf. Mach. Learn.*, pp. 3987–3995, PMLR, 2017.

[49] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Eur. Conf. Comput. Vis.*, pp. 139–154, 2018.

[50] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 7765–7773, 2018.

[51] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Eur. Conf. Comput. Vis.*, pp. 67–82, 2018.

[52] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *Int. Conf. Mach. Learn.*, pp. 4548–4557, PMLR, 2018.

[53] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3366–3375, 2017.

[54] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 651–663, 2018.

[55] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[56] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *AAAI Conf. Artif. Intell.*, vol. 32, pp. 3302–3309, 2018.

[57] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. Dokania, P. Torr, and M. Ranzato, "Continual learning with tiny episodic memories," *arXiv preprint arXiv:1902.10486*, 2019.

[58] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," vol. 32, pp. 348–358, 2019.

[59] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *Proc. IEEE Winter Conf. Appli. Comput. Vis.*, pp. 1131–1140, 2020.

[60] M. Boschini, L. Bonicelli, P. Buzzega, A. Porrello, and S. Calderara, "Class-incremental continual learning into the extended der-verse," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 5497–5512, 2022.

[61] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Co-transport for class-incremental learning," in *ACM Int. Conf. Multimedia*, pp. 1645–1654, 2021.

[62] K. Zhu, W. Zhai, Y. Cao, J. Luo, and Z.-J. Zha, "Self-sustaining representation expansion for non-exemplar class-incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9296–9305, 2022.

[63] Y. Wang, Z. Huang, and X. Hong, "S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning," *Adv. Neural Inform. Process. Syst.*, vol. 35, pp. 5682–5695, 2022.

[64] M. J. Mirza, M. Masana, H. Possegger, and H. Bischof, "An efficient domain-incremental learning approach to drive in all weather conditions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3001–3011, 2022.

[65] H. Shi and H. Wang, "A unified approach to domain incremental learning with memory: Theory and algorithm," in *Adv. Neural Inform. Process. Syst.*, vol. 36, 2024.

[66] J. He, R. Mao, Z. Shao, and F. Zhu, "Incremental learning in online scenario," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13926–13935, 2020.

[67] A. Chrysakis and M.-F. Moens, "Online continual learning from imbalanced data," in *Int. Conf. Mach. Learn.*, pp. 1952–1961, PMLR, 2020.

[68] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Adv. Neural Inform. Process. Syst.*, vol. 32, pp. 11816–11825, 2019.

[69] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 12183–12192, 2020.

[70] S. Dong, X. Hong, X. Tao, X. Chang, X. Wei, and Y. Gong, "Few-shot class-incremental learning via relation knowledge distillation," in *AAAI Conf. Artif. Intell.*, vol. 35, pp. 1255–1263, 2021.

[71] A. Cheraghian, S. Rahman, P. Fang, S. K. Roy, L. Petersson, and M. Harandi, "Semantic-aware knowledge distillation for few-shot class-incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 2534–2543, 2021.

[72] X. Liu, Y.-S. Hu, X.-S. Cao, A. D. Bagdanov, K. Li, and M.-M. Cheng, "Long-tailed class incremental learning," in *Eur. Conf. Comput. Vis.*, pp. 495–512, Springer, 2022.

[73] J. He, L. Lin, J. Ma, H. A. Eicher-Miller, and F. Zhu, "Long-tailed continual learning for visual food recognition," *arXiv preprint arXiv:2307.00183*, 2023.

[74] P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, and R. Chellappa, "Learning without memorizing," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 5138–5146, 2019.

[75] E. Fini, S. Lathuiliere, E. Sangineto, M. Nabi, and E. Ricci, "Online continual learning under extreme memory constraints," in *Eur. Conf. Comput. Vis.*, pp. 720–735, Springer, 2020.

[76] X. Tao, X. Chang, X. Hong, X. Wei, and Y. Gong, "Topology-preserving class-incremental learning," in *Eur. Conf. Comput. Vis.*, pp. 254–270, Springer, 2020.

[77] N. Asadi, M. Davari, S. Mudur, R. Aljundi, and E. Belilovsky, "Prototype-sample relation distillation: towards replay-free continual learning," in *Int. Conf. Mach. Learn.*, pp. 1093–1106, PMLR, 2023.

[78] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Eur. Conf. Comput. Vis.*, pp. 233–248, 2018.

[79] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Lifelong learning via progressive distillation and retrospection," in *Eur. Conf. Comput. Vis.*, pp. 437–452, 2018.

[80] K. Lee, K. Lee, J. Shin, and H. Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," in *Int. Conf. Comput. Vis.*, pp. 312–321, 2019.

[81] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *Eur. Conf. Comput. Vis.*, pp. 86–102, Springer, 2020.

[82] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 16071–16080, 2022.

[83] H. Cha, J. Lee, and J. Shin, "Co2l: Contrastive continual learning," in *Int. Conf. Comput. Vis.*, pp. 9516–9525, 2021.

[84] C. Simon, P. Koniusz, and M. Harandi, "On learning the geodesic path for incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1591–1600, 2021.

[85] Y. Liu, X. Hong, X. Tao, S. Dong, J. Shi, and Y. Gong, "Model behavior preserving for class-incremental learning," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[86] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," in *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 15920–15930, 2020.

[87] J. Li, Z. Ji, G. Wang, Q. Wang, and F. Gao, "Learning from students: Online contrastive distillation network for general continual learning," in *Int. Joint Conf. Artif. Intell.*, pp. 3215–3221, 2022.

[88] C. Wu, L. Herranz, X. Liu, J. Van De Weijer, B. Raducanu, *et al.*, "Memory replay gans: Learning to generate new categories without forgetting," in *Adv. Neural Inform. Process. Syst.*, vol. 31, pp. 5966–5976, 2018.

[89] W. Hu, Z. Lin, B. Liu, C. Tao, Z. T. Tao, D. Zhao, J. Ma, and R. Yan, "Overcoming catastrophic forgetting for continual learning via model adaptation," in *Int. Conf. Learn. Represent.*, 2019.

[90] J. Smith, Y.-C. Hsu, J. Balloch, Y. Shen, H. Jin, and Z. Kira, "Always be dreaming: A new approach for data-free class-incremental learning," in *Int. Conf. Comput. Vis.*, pp. 9374–9384, 2021.

[91] Q. Gao, C. Zhao, B. Ghanem, and J. Zhang, "R-dfcil: Relation-guided representation learning for data-free class incremental learning," in *Eur. Conf. Comput. Vis.*, pp. 423–439, Springer, 2022.

[92] M. PourKeshavarzi, G. Zhao, and M. Sabokrou, "Looking back on learned experiences for class/task incremental learning," in *Int. Conf. Learn. Represent.*, 2021.

[93] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 374–382, 2019.

[94] K. Javed and F. Shafait, "Revisiting distillation and incremental classifier learning," in *Asian Conf. Comput. Vis.*, pp. 3–17, Springer, 2019.

[95] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13208–13217, 2020.

[96] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 831–839, 2019.

[97] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, and T. Moon, "Ss-il: Separated softmax for incremental learning," in *Int. Conf. Comput. Vis.*, pp. 844–853, 2021.

[98] Z. Wang, L. Liu, Y. Duan, Y. Kong, and D. Tao, "Continual learning with lifelong vision transformer," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 171–181, 2022.

[99] X. Chen and X. Chang, "Dynamic residual classifier for class incremental learning," in *Int. Conf. Comput. Vis.*, pp. 18743–18752, 2023.

[100] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, and Q. Sun, "Mnemonics training: Multi-class incremental learning without forgetting," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 12245–12254, 2020.

[101] R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy, "Gcr: Gradient coreset based replay buffer selection for continual learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 99–108, 2022.

[102] J. Rajasegaran, M. Hayat, S. H. Khan, F. S. Khan, and L. Shao, "Random path selection for continual learning," in *Adv. Neural Inform. Process. Syst.*, vol. 32, pp. 12648–12658, 2019.

[103] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "Dytox: Transformers for continual learning with dynamic token expansion," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9285–9295, 2022.

[104] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *Eur. Conf. Comput. Vis.*, pp. 398–414, Springer, 2022.

[105] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A. D. Bagdanov, S. Jui, and J. v. de Weijer, "Generative feature replay for class-incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 226–227, 2020.

[106] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, "Prototype augmentation and self-supervision for incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 5871–5880, 2021.

[107] F. Zhu, Z. Cheng, X.-Y. Zhang, and C.-l. Liu, "Class-incremental learning via dual augmentation," in *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 14306–14318, 2021.

[108] S. Roy, M. Liu, Z. Zhong, N. Sebe, and E. Ricci, "Class-incremental novel class discovery," in *Eur. Conf. Comput. Vis.*, pp. 317–333, Springer, 2022.

[109] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[110] G. Petit, A. Popescu, H. Schindler, D. Picard, and B. Delezoide, "Fetril: Feature translation for exemplar-free class-incremental learning," in *Proc. IEEE Winter Conf. Appli. Comput. Vis.*, pp. 3911–3920, January 2023.

[111] W. Shi and M. Ye, "Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning," in *Int. Conf. Comput. Vis.*, pp. 1772–1781, 2023.

[112] M. Toldo and M. Ozay, "Bring evanescent representations to life in lifelong class incremental learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 16732–16741, 2022.

[113] A. Iscen, J. Zhang, S. Lazebnik, and C. Schmid, "Memory-efficient incremental learning through feature adaptation," in *Eur. Conf. Comput. Vis.*, pp. 699–715, Springer, 2020.

[114] G. M. Van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Communications*, vol. 11, no. 1, p. 4069, 2020.

[115] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 8715–8724, 2020.

[116] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9729–9738, 2020.

[117] Y. Lu, M. Wang, and W. Deng, "Augmented geometric distillation for data-free incremental person reid," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 7329–7338, 2022.

[118] F. Lyu, S. Wang, W. Feng, Z. Ye, F. Hu, and S. Wang, "Multi-domain multi-task rehearsal for lifelong learning," in *AAAI Conf. Artif. Intell.*, vol. 35, pp. 8819–8827, 2021.

[119] M. Boschini, L. Bonicelli, A. Porrello, G. Bellitto, M. Pennisi, S. Palazzo, C. Spampinato, and S. Calderara, "Transfer without forgetting," in *Eur. Conf. Comput. Vis.*, pp. 692–709, Springer, 2022.

[120] Q. Pham, C. Liu, and S. Hoi, "Dualnet: Continual learning, fast and slow," in *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 16131–16144, 2021.

[121] J. Xie, S. Yan, and X. He, "General incremental learning with domain-aware categorical representations," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 14351–14360, 2022.

[122] Y.-M. Tang, Y.-X. Peng, and W.-S. Zheng, "Learning to imagine: Diversify memory for incremental learning using unlabeled data," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9549–9558, 2022.

[123] Q. Yan, D. Gong, Y. Liu, A. van den Hengel, and J. Q. Shi, "Learning bayesian sparse networks with full experience replay for continual learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 109–118, 2022.

[124] K. Joseph, S. Paul, G. Aggarwal, S. Biswas, P. Rai, K. Han, and V. N. Balasubramanian, "Novel class discovery without forgetting," in *Eur. Conf. Comput. Vis.*, pp. 570–586, Springer, 2022.

[125] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 18661–18673, 2020.

[126] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong gan: Continual learning for conditional image generation," in *Int. Conf. Comput. Vis.*, pp. 2759–2768, 2019.

[127] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Int. Conf. Comput. Vis.*, pp. 618–626, 2017.

[128] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. Learn. Represent.*, 2014.

[129] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[130] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, 2008.

[131] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 248–255, IEEE, 2009.

[132] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *Stanford CS 231N*, vol. 7, no. 7, p. 3, 2015.

[133] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Adv. Neural Inform. Process. Syst.*, vol. 30, pp. 6467–6476, 2017.

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

[135] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 770–778, 2016.

[136] E. Belouadah and A. Popescu, "Il2m: Class incremental learning with dual memory," in *Int. Conf. Comput. Vis.*, pp. 583–592, 2019.

[137] A. Prabhu, H. A. Al Kader Hammoud, P. K. Dokania, P. H. Torr, S.-N. Lim, B. Ghanem, and A. Bibi, "Computationally budgeted continual learning: What does matter?," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3698–3707, 2023.

[138] C. Peng, K. Zhao, and B. C. Lovell, "Faster ilod: Incremental learning for object detectors based on faster rcnn," *Pattern recognition letters*, vol. 140, pp. 109–115, 2020.

[139] Y. Liu, Y. Cong, D. Goswami, X. Liu, and J. van de Weijer, "Augmented box replay: Overcoming foreground shift for incremental object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11367–11377, 2023.

[140] M. H. Phan, S. L. Phung, L. Tran-Thanh, A. Bouzerdoum, *et al.*, "Class similarity weighted knowledge distillation for continual semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 16866–16875, 2022.

[141] Z. Yang, R. Li, E. Ling, C. Zhang, Y. Wang, D. Huang, K. T. Ma, M. Hur, and G. Lin, "Label-guided knowledge distillation for continual semantic segmentation on 2d images and 3d point clouds," in *Int. Conf. Comput. Vis.*, pp. 18601–18612, 2023.

[142] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, "Plop: Learning without forgetting for continual semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4040–4050, 2021.

[143] Y. Jin, J. Wang, and D. Lin, "Multi-level logit distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 24276–24285, 2023.

[144] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 11953–11962, 2022.

[145] Y. Niu, L. Chen, C. Zhou, and H. Zhang, "Respecting transfer gap in knowledge distillation," in *Adv. Neural Inform. Process. Syst.*, vol. 35, pp. 21933–21947, 2022.
[146] X. Liu, H. Yang, A. Ravichandran, R. Bhotika, and S. Soatto, "Multi-task incremental learning for object detection," *arXiv preprint arXiv:2002.05347*, 2020.
[147] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Int. Conf. Comput. Vis.*, pp. 3400–3409, 2017.
[148] B. Yang, X. Deng, H. Shi, C. Li, G. Zhang, H. Xu, S. Zhao, L. Lin, and X. Liang, "Continual object detection via prototypical task correlation guided gating mechanism," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9255–9264, 2022.
[149] J. Sun, S. Wang, J. Zhang, and C. Zong, "Distill and replay for continual language learning," in *Proceedings of the 28th international conference on computational linguistics*, pp. 3569–3579, 2020.
[150] C. Qin and S. Joty, "Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5," *arXiv preprint arXiv:2110.07298*, 2021.
[151] Y. Cao, H.-R. Wei, B. Chen, and X. Wan, "Continual learning for neural machine translation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3964–3974, 2021.
[152] C. Shao and Y. Feng, "Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation," *arXiv preprint arXiv:2203.03910*, 2022.
[153] Q. Liu, X. Yu, S. He, K. Liu, and J. Zhao, "Lifelong intent detection via multi-strategy rebalancing," *arXiv preprint arXiv:2108.04445*, 2021.
[154] G. Li, Y. Zhai, Q. Chen, X. Gao, J. Zhang, and Y. Zhang, "Continual few-shot intent detection," in *Proceedings of the 29th international conference on computational linguistics*, pp. 333–343, 2022.
[155] N. Monaikul, G. Castellucci, S. Filice, and O. Rokhlenko, "Continual learning for named entity recognition," in *AAAI Conf. Artif. Intell.*, vol. 35, pp. 13570–13577, 2021.
[156] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 139–149, 2022.
[157] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *Eur. Conf. Comput. Vis.*, pp. 631–648, Springer, 2022.
[158] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira, "Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 11909–11919, 2023.
[159] Y. Wang, Z. Huang, and X. Hong, "S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning," in *Adv. Neural Inform. Process. Syst.*, vol. 35, pp. 5682–5695, 2022.
[160] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *Int. Conf. Mach. Learn.*, pp. 2790–2799, PMLR, 2019.
[161] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.
[162] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.
[163] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models," in *Int. Conf. Learn. Represent.*, 2021.
[164] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, "A survey on knowledge distillation of large language models," *arXiv preprint arXiv:2402.13116*, 2024.
[165] H. Shi, Z. Xu, H. Wang, W. Qin, W. Wang, Y. Wang, and H. Wang, "Continual learning of large language models: A comprehensive survey," *arXiv preprint arXiv:2404.16789*, 2024.
[166] Z. Hao, J. Guo, K. Han, Y. Tang, H. Hu, Y. Wang, and C. Xu, "One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation," in *Adv. Neural Inform. Process. Syst.*, vol. 36, 2024.
[167] Z. Xue, S. Ren, Z. Gao, and H. Zhao, "Multimodal knowledge expansion," in *Int. Conf. Comput. Vis.*, pp. 854–863, 2021.
[168] Z. Xue, Z. Gao, S. Ren, and H. Zhao, "The modality focusing hypothesis: Towards understanding crossmodal knowledge distillation," *arXiv preprint arXiv:2206.06487*, 2022.

Songze Li received his B.S. degree from the School of Software, Harbin Institute of Technology in 2016. He is currently pursuing the Ph.D. degree in software engineering at Harbin Institute of Technology (HIT), China. His research interests include continual learning, computer vision and edge computing.

Tonghua Su (Ph. D) is the associate professor at Harbin Institute of Technology (HIT). His research interests include large-scale pattern recognition and heterogeneous computing architecture, especially the deep learning driven agents. He had released the first Chinese handwritten text database, which was used by more than 200 universities or institutes. He had initialized the segmentation-free strategy for Chinese handwriting recognition and now evolved as end-to-end strategy. He had won two academic competitions with1st place. He had been honored as the best GPU educator for four years over China. He has published five monographs and translated ten books.

Xuyao Zhang (Senior Member, IEEE) received the B.S. degree in computational mathematics from Wuhan University, Wuhan, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013. He was a Visiting Researcher with the Center for Pattern Recognition and Machine Intelligence (CENPARMI) of Concordia University in 2012. From March 2015 to March 2016, he was a Visiting Scholar with the Montreal Institute for Learning Algorithms (MILA) at the University of Montreal. He is currently a Professor with the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Institute of Automation of Chinese Academy of Sciences, Beijing, China. His research interests include pattern recognition, machine learning and handwriting recognition.

Zhongjie Wang is a professor at Faculty of Computing, Harbin Institute of Technology (HIT). He received the Ph.D. degree in computer science from Harbin Institute of Technology in 2006. His research interests include services computing, mobile and social networking services, and software architecture. He is the author of more than 80 publications. He is a member of the IEEE.