Universiteit Leiden
The Netherlands

Open-world continual learning via knowledge transfer

Li, Y.

Chapter 4

Learning to Prompt Knowledge Transfer for Open-world Continual Learning

As discussed in Chapter 3, open-world continual learning (OWCL) extends traditional continual learning by embracing the open-world assumption, an appealing yet highly challenging paradigm. In this chapter, we begin by presenting Figure 4.1 and Figure 4.2 to visually illustrate the open-world continual learning setting and its key distinctions. We then further introduce the core challenges associated with open-world continual learning, setting the stage for the methods and solutions explored throughout the chapter.

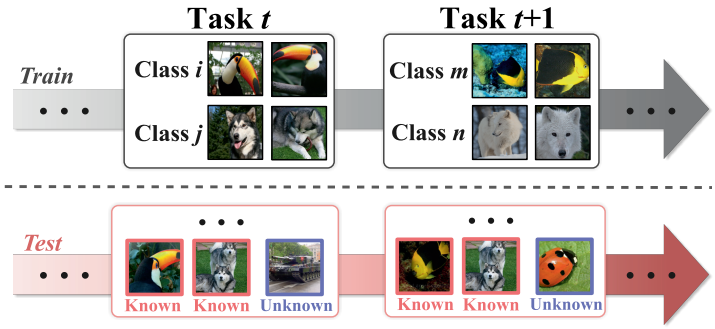


Figure 4.1: A simple illustration of OWCL.

As illustrated in Figure 4.1, each new task incrementally introduces new classes (e.g., *class m* and *class n* in task $t+1$), while the test sets for both task t and task $t+1$ contain open classes (highlighted in blue) that were not observed during training. This reflects the fundamental challenge of open-world continual learning, where models must cope with both continual task acquisition and the presence of unknown categories unseen during training.

To further highlight the limitations of existing continual learning methods under open-world continual learning settings, we conducted a series of preliminary experiments. These experiments aim to uncover the performance bottlenecks that arise when conventional continual learning models are extended to handle open-world scenarios. As shown in Figure 4.2, we evaluate the evolving model performance on the test sets of all tasks after sequentially learning each new task. This evaluation compares a representative continual learning baseline with our proposed approach.

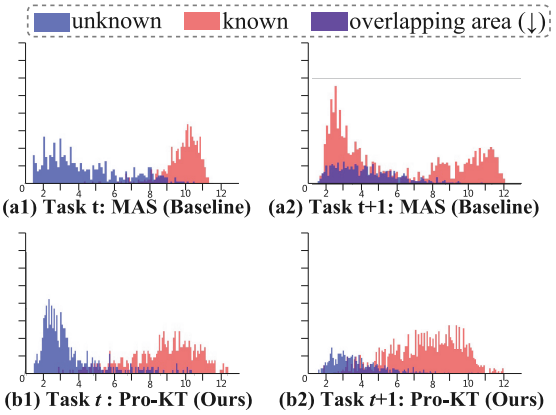


Figure 4.2: Distributions of the outputted logits scores of test samples, using MAS (a typical continual learning baseline) and the proposed *Pro-KT* on Task 1 and Task 2, respectively. The x-axis is the unscaled class-maximum logits of samples, and the y-axis shows the number of samples.

In Figure 4.2 (a1) and (b1), both MAS [96] and our proposed method demonstrate an ability to distinguish known from unknown classes in the previous task (t). However, in the current task ($t+1$), MAS fails to maintain this separation. As seen in Figure 4.2 (a2), the logit distributions for unknown and known samples significantly overlap, indicating MAS's inability to detect open-set entities, resulting in degraded performance. In contrast, our method *Pro-KT*, depicted in Figure 4.2 (b2), exhibits better separation between knowns and unknowns, with reduced overlap in the logit distributions.

These observations form the basis of our main motivation: to address the research question **RQ1** introduced in chapter 1 by developing an open-world continual learning model that not only accumulates knowledge over time but also uses it to accurately distinguish between known and unknown categories. Specifically, our goal is to expand the open-set boundary to improve recognition of unfamiliar instances. After learning each new task, the previously unseen classes are incorporated into the set of knowns, progressively refining the model's understanding of the environment [97].

Challenges and Our Approach: Open-world continual learning presents several core challenges. (1) The first is the issue of *knowledge stability*, which arises when new tasks diverge significantly from previously encountered ones. In such cases, newly acquired knowledge may conflict with earlier representations, risking degradation of past learning. (2) The second challenge is *knowledge plasticity*, which requires the model not only to incrementally learn new tasks but also to reliably detect open or unknown classes during testing, and ideally, to integrate those unknowns into the set of knowns. The tension between stability and plasticity is fundamental to open-world continual learning, as it governs the model's ability to retain prior knowledge while adapting to novel information. Effectively balancing these forces is crucial for ensuring both robustness and adaptability in evolving environments.

To address these challenges, we propose a new framework: **Pro**mpt-enhanced **K**nowledge **T**ransfer (*Pro-KT*). *Pro-KT* represents a significant advancement in prompt-based continual learning. It features a modular prompt bank to facilitate structured knowledge transfer and introduces two adaptive strategies for open-set boundary modeling to manage the dynamic nature of open-world learning.

To mitigate Challenge (1) (knowledge stability), *Pro-KT* constructs a prompt bank that encapsulates both general and task-specific knowledge. These prompts act as modular, instructional units that guide the model's behavior across tasks. By selectively activating relevant prompts, the model achieves effective, scalable transfer without overwriting prior knowledge.

To address Challenge (2) (knowledge plasticity) and answer research question RQ1 from Chapter 1.3, *Pro-KT* introduces two adaptive thresholding mechanisms for open-set recognition. These strategies allow the model to dynamically refine decision boundaries as it accumulates new knowledge, supporting continual integration of novel classes while preserving discrimination capabilities.

Our main contributions and advantages of *Pro-KT* are summarized as follows:

• *Prompt-based, rehearsal-free knowledge transfer paradigm.* We introduce a novel prompt bank designed for open-world continual learning, enabling structured representation and transfer of knowledge without relying on data replay. This decouples learning from memory-intensive rehearsal and provides a scalable alternative for continual learners.

• *Balanced handling of stability and plasticity via task-generic and task-specific prompts.* The prompt bank encodes both general and specialized task knowledge, enabling flexible and efficient adaptation. Prompts are selected dynamically based on the task context, allowing the model to generalize across tasks while remaining sensitive to new information.

• *Adaptive open-set recognition through dynamic boundary modeling.* We develop two strategies for task-aware thresholding, allowing the model to distinguish known from

unknown classes throughout continual learning. These adaptive boundaries evolve with the learning process, supporting seamless incorporation of new concepts.

• *Empirical effectiveness and robustness across real-world benchmarks.* Extensive experiments on two real-world open-world continual learning benchmark suites demonstrate that *Pro-KT* consistently outperforms strong baselines across a wide range of tasks. Qualitative visualizations and case studies confirm the model's practical utility in handling open-set and dynamic learning conditions.

This chapter is based on the following publication [30]:

- **Li, Y.**, Yang, X., Wang, H., Wang, X., and Li, T.: Learning to Prompt Knowledge Transfer for Open-World Continual Learning. In: The 38th Annual AAAI Conference on Artificial Intelligence. AAAI '24 (2024) 38(12), 13700-13708.

4.1 Objectives of OWCL

Recall from Definition 4 in Chapter 3 that the *Open-world Continual Learning* problem arises when a learner is exposed to a (potentially infinite) sequence of tasks and must handle the emergence of novel, previously unseen objects during testing. To prevent misclassification, such novel/open instances should be identified as unknowns rather than being erroneously assigned to known categories. Subsequently, when ground-truth labels for these unknowns become available in future tasks' training sets, the learner should evolve them into known classes and integrate them into its knowledge base accordingly.

Furthermore, the learner must learn continuously and incrementally on the job without access to training data from previous tasks. In other words, it must perform *incremental learning without forgetting* [30], thereby minimizing the incremental prediction error while retaining knowledge from earlier tasks. Notably, retraining the model from scratch is not an option.

In this chapter, we formally refine the definition and objectives of open-world continual learning. The notation used throughout this chapter is summarized in Table 4.1. It is worth noting that, unless otherwise specified, all superscripts in this chapter denote task identifiers.

Definition 7 ***Objectives of OWCL.*** *open-world continual learning learns from a sequence of tasks,* $1, ..., t, ...$ *Each task* t *provides a training dataset* $\mathbf{D}_{tr}^t = \{(x_i^t, y_i^t)_{i=1}^n\}$*, where* n *is the number of training samples in task* t*,* $x_i^t \in \mathbf{X}^t$ *is an input instance, and* $y_i^t \in \mathbf{Y}^t$ *is its corresponding class label. Here,* $\mathbf{Y}^t$ *denotes the set of all classes in task* t*. The label sets are mutually disjoint, i.e.,* $\left(\mathbf{Y}^t \cap \mathbf{Y}^{\hat{t}} = \emptyset,\ \forall t \neq \hat{t}\right)$*, and the cumulative set of all training labels is given by* $\bigcup_{t=1}^{T} \mathbf{Y}^t = \mathbf{Y}_{tr}$*. Similarly, we denote by* $\mathbf{Y}_{te}$ *the cumulative set of all labels at the testing phase.*

Table 4.1: Notation used in this chapter.

Notation	Explanation
$(\mathbf{P}, \mathbf{K})$	Prompt bank
N	Total # of tasks
M	Total # of prompts learned from each task
$(\mathbf{k}^t, \mathbf{p}^t)$	The set of key-query pair of t
(k_i^j, p_i^j)	The i key-query pair in task j
L_P	The token length of each prompt
D_e	The embedding dimension
$\mathbf{Q}_x$	The query function
$\mathbf{h}_i^t$	The embedding of ${x_i}^t$
$\mathbf{P}_{S_K}$	A subset of the prompt bank
$\mathbf{h'}_i^t$	The input embedding extended by $\mathbf{P}_{S_K}$
f_{pr}	Pre-trained backbone
f_{θ^t}	Trainable classifier for t
$prob_i^t$	Maximum unscaled softmax entropy of ${x_i}^t$
$\lvert\mathbf{X}^t\rvert$	Total # of task t's training samples
r	A deviation degree
λ	A trade-off degree
$[\mu^1, ..., \mu^t, ..., \mu^N]$	All learned open detection thresholds.

Open-world continual learning accommodates the emergence of previously unseen or open classes during testing, meaning that we do not require $\mathbf{Y}_{te}$ *to be a subset of* $\mathbf{Y}_{tr}$*. Therefore, the goal of open-world continual learning is to learn a single predictive function* $f_{\theta_t} : \mathbf{X} \to \mathbf{Y}$ *after learning task* t *that can:* ***1)*** *detect and reject open/unknown test instances, and* ***2)*** *correctly classify known test instances—while retaining knowledge across tasks and mitigating catastrophic forgetting.*

4.2 Methodology

Figure 4.3 presents the overall architecture of the proposed *Pro-KT* framework, which operates in two stages: a *training* phase and a *testing* phase.

As previously discussed, traditional continual learning approaches are typically based on the closed-world assumption, an unrealistic constraint in practical settings, where previously unseen and unknown classes frequently emerge. Although recent efforts have explored this problem under the open-world continual learning paradigm, the task remains inherently challenging. Existing methods often misclassify novel inputs, accumulate fragmented or incoherent knowledge, or even reinforce incorrect patterns, while largely neglecting the potential of explicit knowledge transfer mechanisms.

To overcome these challenges, *Pro-KT* introduces a prompt-based framework designed to encode, transfer, and evolve task-specific and task-generic knowledge in an open-world setting. Unlike rehearsal-based methods, *Pro-KT* avoids storing past

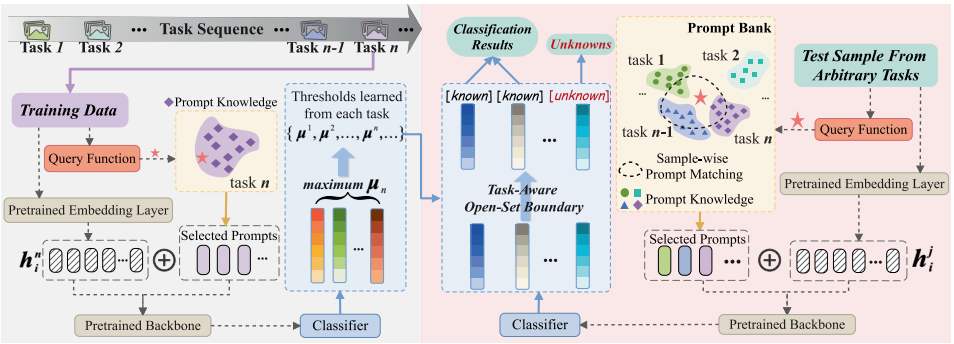


Figure 4.3: Overview of the proposed *Pro-KT* framework. Left: Training phase. Right: Testing phase. During training, input samples are first projected using a query function. *Pro-KT* learns a new set of prompts and stores them in a prompt bank. Each sample is then enriched with selected prompts, and the resulting prompt-enhanced embeddings are processed by a frozen pre-trained backbone. The final representations are passed to a trainable classifier. Additionally, an adaptive threshold for a task-aware open-set boundary is learned based on the softmax entropy of these enhanced representations. In the *testing phase*, test samples are similarly projected and matched with prompts from the prompt bank using a sample-wise selection strategy, enabling transfer of both task-specific and generic knowledge. The enriched embeddings are fed into the unified classifier to obtain unscaled logit scores. Finally, *Pro-KT* classifies each sample as either [*known*] or [*unknown*], using the learned adaptive threshold.

data by maintaining a structured prompt bank that supports modular, efficient, and scalable knowledge transfer across tasks.

With this motivation and problem setting in mind, we now detail the core components of the *Pro-KT* framework, including its prompting strategy, training objectives, and overall learning procedure.

4.2.1 Prompt Bank

We propose leveraging a prompt bank as the central mechanism for knowledge transfer in open-world continual learning. Specifically, prompts are used to encode both task-generic and task-specific knowledge, which are retained in a dedicated prompt bank. These stored prompts are later transferred to new tasks, enabling effective reuse of prior knowledge and supporting long-term knowledge stability across the continual learning process.

Given the current task n with its associated training data $D_{tr}^n = (\mathbf{X}^n, \mathbf{Y}^n)$, the learner derives a set of prompts $(p_1^n, ..., p_i^n, ...)$ tailored to this task. These task-specific prompts are then stored alongside prompts from previous tasks in the prompt

bank:

$$\begin{aligned}\mathbf{P} &= \bigcap_{n=1}^{N}(\mathbf{p}^1, ..., \mathbf{p}^n, ..., \mathbf{p}^N) \\ &= \{(p_1^1, p_2^1, ..., p_M^1), ..., \ (p_1^n, p_2^n, ..., p_M^n), ..., (p_1^N, p_2^N, ..., p_M^N)\},\end{aligned} \tag{4.1}$$

where N is the total number of tasks and M denotes the number of prompts learned per task. Each prompt $p_i^j \in \mathbb{R}^{L_p \times D_e \times NM}$ represents the i-th prompt from task j, where L_p is the prompt token length and D_e is the embedding dimension. This structured prompt bank promotes knowledge transfer by capturing both specific and generalizable representations from each task.

A key challenge, however, lies in how to encode and selectively retrieve the most relevant prompting knowledge. Inspired by the key-value query mechanism introduced in L2P [32], we associate each prompt with a learnable key. For a given task n, we construct key-value pairs to support sample-wise prompt matching as follows:

$$(\mathbf{k}^n, \mathbf{p}^n) = (k_1^n, p_1^n), ..., (k_M^n, p_M^n), \tag{4.2}$$

where $\mathbf{p}^n = p_m^n M$ is the set of prompts and $\mathbf{k}^n = k_m^n M$ is the corresponding set of keys. Upon learning task n, the prompt bank maintains the full set of learned key-prompt pairs as:

$$(\mathbf{K}, \mathbf{P}) = \{(\mathbf{k}^1, \mathbf{p}^1), (\mathbf{k}^2, \mathbf{p}^2), ..., (\mathbf{k}^N, \mathbf{p}^N)\}. \tag{4.3}$$

To facilitate more flexible and automatic knowledge transfer, we introduce a sample-wise prompt matching mechanism, where each input sample autonomously selects the most relevant prompts from the prompt bank to serve as auxiliary knowledge. Given an input ${x_i}^n$ from task n, we first project it into a feature space $\mathbb{R}^{D_e}$ using a pre-trained projection layer $\mathbf{Q}_x$:

$$\mathbf{h}_i^n = \mathbf{Q}_x \cdot {x_i}^n. \tag{4.4}$$

This projected embedding $\mathbf{h}_i^n$ then acts as a query to retrieve the top-K most relevant prompts by matching against all stored keys in the prompt bank. Specifically, we solve the following optimization problem to identify the K keys with the smallest distance (measured using cosine similarity in this work):

$$\underset{[1,K]}{\operatorname{argmin}} \sum_{j=1}^{N} dis(\mathbf{h}_i^n, \mathbf{k}^j), \text{with } \mathbf{k}^j = \{k_m^j\}_M. \tag{4.5}$$

This procedure enables each input to retrieve the most relevant prompts purely based on semantic similarity, without requiring access to any task identity. Once selected, the input embedding $\mathbf{h}i^n$ is augmented with the retrieved subset of prompts

$\mathbf{P}S_K \subset \mathbf{P}$ from the prompt bank:

$$\mathbf{h}'^n_i = \mathbf{h}^n_i \oplus \mathbf{P}_{S_K}, \tag{4.6}$$

where $\oplus$ denotes dimension-wise concatenation, and the subset size K is smaller than the total number of prompts M per task.

To handle the presence of unknown classes during testing, these enriched embeddings are subsequently used to learn an adaptive detection boundary, which allows the model to distinguish between known and unknown samples on a per-task basis.

4.2.2 Task-Aware Open-Set Boundary

In this section, we describe how to establish an adaptive boundary between known and unknown samples during the open-world continual learning process. To this end, we propose two adaptive thresholding schemes for open-set detection, aimed at determining this boundary in a task-aware manner.

During training, the prompt-based knowledge stored in the prompt bank is derived from the labeled training data of each task. Crucially, we leverage the information learned from known classes to help detect unknown samples. Specifically, we propose learning a threshold based on the softmax entropy of prompt-enhanced embeddings, using this known-class information to estimate a decision boundary that separates knowns from unknowns.

After completing the training for task n, each prompt-enhanced training sample $\mathbf{h}'i^n$ is passed through a pre-trained backbone fpr, and the resulting representation is fed into a trainable classifier f_{θ^n}:

$$prob^n_i = f_{\theta^n}(f_{pr}(\mathbf{h}'^n_i)), \tag{4.7}$$

where $prob^n_i$ represents the maximum value of the unscaled softmax entropy corresponding to the training sample $\mathbf{h}'^n_i$.

To derive the open-set detection threshold μ, we compute the average softmax entropy across all training samples $\{prob^n_1, ..., prob^n_i, ...\}$ from task n and scale it using a deviation factor r:

$$\mu^n = r \cdot \frac{1}{|\mathbf{X}^n|} \sum_{i=1}^{|\mathbf{X}^n|} prob^n_i, \tag{4.8}$$

where $|\mathbf{X}^n|$ denotes the number of training samples, and r is a hyperparameter that controls the sensitivity of the threshold.

Once computed, the threshold μ^n for task n is stored alongside thresholds from previous tasks. This collection $[\mu^1, \ldots, \mu^n]$ allows the model to dynamically retrieve a suitable threshold during inference in the testing phase.

In the testing phase, when evaluating whether a test sample x_i^j from any task T^j belongs to a known or unknown class, we use two simple yet effective strategies to select the most appropriate threshold from the stored set. These strategies guide the open-set detection mechanism and help maintain robust performance across tasks.

We consider two testing scenarios for open-set detection. In the first scenario, task identifiers (task IDs) are available at test time. This allows the model to directly use the threshold associated with the given task ID j for open-set classification:

$$\begin{aligned} x_i^j : f_{\theta^n}(\mathbf{h}'^j_i) \leq \mu^j \rightarrow [unknown], \\ \text{otherwise, } \rightarrow [known]. \end{aligned} \tag{4.9}$$

In the second scenario, task IDs are unavailable during testing—a more realistic and challenging setting for open-world continual learning. As the model learns tasks incrementally, the prompt bank accumulates knowledge from all previously seen tasks. In this case, the model selects the most recently learned threshold μ^n learned from the current task n for the open-set detection boundary:

$$\begin{aligned} x_i^j : f_{\theta^n}(\mathbf{h}'^j_i) \leq \mu^n \rightarrow [unknown], \\ \text{otherwise, } \rightarrow [known]. \end{aligned} \tag{4.10}$$

After applying this decision rule, if a sample is classified as an unknown object, it is labeled as $[unknown]$ and set aside for potential inclusion in future tasks. However, if the sample is classified as belonging to one of the known classes, the model proceeds to assign it the appropriate class label from the set of previously learned categories.

4.2.3 The Overall Objective and Algorithms

During the training phase of task n, the prompt bank is updated by learning M new prompts along with their associated keys. The prompt bank $(\mathbf{K}, \mathbf{P})$ retains all the prompt knowledge accumulated from tasks T^1 through n. For each known training sample, the model retrieves the top-K most relevant prompts using the key-query matching described in Equation 4.5, and extends the corresponding embedding with auxiliary prompt knowledge as defined in Equation 4.6. At the same time, the open-set detection boundary is refined through the adaptive threshold-setting strategies proposed earlier.

The overall training objective for task n can be formally defined as:

$$\min_{\theta^n, (\mathbf{k}^n, \mathbf{p}^n)} \{\mathcal{L}(f_{\theta^n}(f_{pr}(\mathbf{h}'^n), \mathbf{y}^n) + \lambda \sum_{i=0}^{M} dis(\mathbf{h}^n, k_i^n)\}, \tag{4.11}$$

where $\mathbf{h}^n$ denotes the projected embeddings of the training samples. The first term represents the standard softmax cross-entropy loss, promoting accurate classification of known samples. The second term is a regularization loss that encourages the selected prompt keys to remain close to the query embeddings, thus reinforcing meaningful key-prompt associations. The hyperparameter λ controls the trade-off between classification performance and prompt alignment.

As previously emphasized, during the training of task n, only the prompt set p^n is updated. In contrast, during testing, the model selects prompts from the entire prompt bank $\mathbf{P}$ to perform open-world prediction.

To provide a clearer understanding of the proposed *Pro-KT* framework, we present the training procedure in algorithm 1 and the testing procedure in algorithm 2, respectively.

Algorithm 1: The Training Process of *Pro-KT*

Input: training set $D_{tr}^n = \{(\mathbf{X}^n, \mathbf{Y}^n)\}$, a prompt bank $(\mathbf{K}, \mathbf{P}) = \{(\mathbf{k}^n, \mathbf{p}^n)\}_{n=1}^N$, trainable classification head of the n-th task f_{θ^n}, a pre-trained Backbone f_{pr}.
Initialize: $(\mathbf{K}, \mathbf{P})$, θ^n, μ^n.

1: **for** n = 1, 2, ..., N **do**
2: **for** each epoch **do**
3: **for** (x_i^n, y_i^n) in $(\mathbf{X}^n, \mathbf{Y}^n)$ **do**
4: Map x_i^n to $\mathbf{h}_i^n$ via Equation 4.4;
5: Calculate the cosine similarity of x_i^n and per key from $(\mathbf{k}^n, \mathbf{p}^n)$;
6: Select $(\mathbf{k}_{S_K}^n, \mathbf{p}_{S_K}^n)$ from $(\mathbf{k}^n, \mathbf{p}^n)$ by top-K algorithm;
7: Extend $\mathbf{h}_i^n$ to $\mathbf{h}'^n_i$ via Equation 4.6;
8: Calculate the prompted feature by $f_{pr}(\mathbf{h}'^n_i)$;
9: Calculate the sample loss via Equation 4.11;
10: **end for**
11: Update θ^n, $(\mathbf{k}^n, \mathbf{p}^n)$ by backpropagation;
12: Calculate the maximum unscaled probability of each training sample via Equation 4.7;
13: Calculate the open-set threshold μ^n via Equation 4.8;
14: updat the set of selectable threshold by appending μ^n to $\{\mu^1, ..., \mu^{(n-1)}\}$;
15: **end for**
16: Update $(\mathbf{K}, \mathbf{P})$.
17: **end for**

The complete training procedure of *Pro-KT* is presented in algorithm 1. For each task $n \in 1, \ldots, N$, the model performs incremental learning using the task-specific labeled dataset $D_{tr}^n = (\mathbf{X}^n, \mathbf{Y}^n)$, in conjunction with the evolving prompt bank $(\mathbf{K}, \mathbf{P})$ and a trainable classification head f_{θ^n}.

During each training epoch, for every input-label pair (x_i^n, y_i^n) from task n, the model begins by projecting the input into an initial embedding $\mathbf{h}_i^n$ (Line 4). It

then computes cosine similarities between this embedding and all keys in the task-specific prompt set $(\mathbf{k}^n, \mathbf{p}^n)$, retrieving the top-$K$ most relevant prompt pairs via a similarity-based matching mechanism (Lines 5–6). These selected prompts are used to enhance the input representation, yielding the prompt-augmented feature vector $\mathbf{h}_i'^n$ (Line 7), which is subsequently processed by the frozen pre-trained backbone f_{pr} to extract the final prompted representation (Line 8).

The model computes the classification loss for each training sample using the predicted logits and the corresponding ground-truth label. This loss is then used to jointly update the classification head θ^n and the task-specific prompt parameters $(\mathbf{k}^n, \mathbf{p}^n)$ via backpropagation (Line 9).

At the end of each epoch, the model estimates a task-specific open-set threshold μ^n by evaluating the maximum unnormalized probability for each training sample. This threshold is then stored and added to the cumulative set of thresholds across tasks to support task-aware open-set recognition during inference at the testing phase (Lines 10–11).

Once all tasks have been trained, the prompt bank $(\mathbf{K}, \mathbf{P})$ is updated to include the newly acquired prompts, thereby consolidating transferable knowledge. This updated prompt bank acts as a transferable memory module that enhances representation learning and supports knowledge transfer in subsequent tasks (Line 13).

Algorithm 2: The Test Process of *Pro-KT*

Given: the prompt bank $\{(\mathbf{k}^n, \mathbf{p}^n)\}_{n=1}^{N}$ trained by current task n , a set of selectable thresholds, trained classification head f_{θ^n}, pre-trained backbone f_{pr}.
Input: a test example x_i^j from T^j.

1: Map x_i^j to $\mathbf{h}_i^j$ via Equation 4.4 and select $(\mathbf{K}_{S_K}, \mathbf{P}_{S_K})$ from $(\mathbf{K}, \mathbf{P})$ via Equation 4.5;
2: Extend $\mathbf{h}$ to $\mathbf{h}_i'^j$ via Equation 4.6 and feed into pre-trained backbone f_{pr};
3: Get the $prob_i^j$ via Equation 4.7;
4: **if** task IDs are available **then**
5: Open detection via Equation 4.9;
6: **else**
7: Open detection: via Equation 4.10;
8: **end if**
9: **if** x_i^j detected as [*known*] **then**
10: output with known classes
11: **end if**

During inference, *Pro-KT* utilizes the learned prompt-based representations to enable accurate and open-aware predictions. The full testing procedure is outlined in algorithm 2.

Given a test sample x_i^j from task T^j, the model performs inference using the learned prompt bank $(\mathbf{k}^n, \mathbf{p}^n)n = 1^N$, the trained task-specific classification heads $f\theta^n$, and a shared pre-trained feature extractor f_{pr}.

The input x_i^j is first projected into its initial embedding $\mathbf{h}_i^j$. Based on cosine similarity with the stored prompt keys, the model selects a top-K subset $(\mathbf{K}_{S_K}, \mathbf{P}_{S_K})$ from the prompt bank to enrich the input representation, resulting in an enhanced embedding $\mathbf{h}_i'^j$ (Line 1). This augmented embedding is then passed through the frozen backbone f_{pr} to obtain the final feature representation (Line 2), which is used to compute the prediction probability vector $prob_i^j$ (Line 3).

To determine whether the test sample corresponds to a known or unknown class, *Pro-KT* performs open-set detection. If task identity is available, the model uses the task-specific threshold strategy defined in Equation 4.9; otherwise, it applies the general strategy described in Equation 4.10 (Lines 4–6).

If the sample is classified as belonging to a known class, *Pro-KT* outputs the final class prediction using the corresponding classifier head f_{θ^j} (Line 7). If the sample is detected as unknown, it is withheld from classification, thus enabling reliable decision-making under the open-world assumption.

4.3 Evaluation

In this section, we present the details of our experimental evaluation to assess the effectiveness of the proposed *Pro-KT* framework in comparison with competitive continual learning and open-set detection baselines. Our experiments are designed to answer the following key questions:

- ***Q1:*** *Does the proposed Pro-KT outperform state-of-the-art continual learning and open-set detection baselines in the open-world continual learning setting?*
- ***Q2:*** *How do the core components of Pro-KT contribute to its overall performance?*
- ***Q3:*** *How sensitive is Pro-KT to different hyperparameter configurations?*

Beyond addressing these primary questions, we also conduct additional experiments to analyze performance across different task sequences and visualize the behavior of the proposed prompt-based knowledge transfer mechanism.

Implementation Details. In our experiments, we simulate the open-world continual learning setting by removing the labels of the classes in the upcoming task (i.e., $T^{(n+1)}$) and treating them as unknowns during testing for the current task (i.e., n). Following standard practice in the continual learning literature, we randomly shuffle the task order five times for both the Split CIFAR-100 benchmark (comprising 10 tasks) and the 5-Datasets benchmark (comprising 5 tasks). All reported results represent the mean over these five randomized task orders.

We implemented both the proposed *Pro-KT* framework and all baseline methods using the PyTorch library, running on a single NVIDIA RTX 3090 GPU. For *Pro-KT*, we adopt the Vision Transformer (ViT) architecture as the backbone network. Input images are resized to 224×224 and normalized to the $[0, 1]$ range. We use the Adam optimizer with a dynamic learning rate schedule, and all models are trained using a batch size of 16 and an initial learning rate of 0.03.

For fair comparison, the hyperparameter configurations for each baseline method are adopted from the optimal settings reported in their respective original papers. On the Split CIFAR-100 benchmark, each task is trained for 20 epochs under the class-incremental learning setting. For the 5-Datasets benchmark, each task is trained for 10 epochs.

As part of our contribution, we also provide faithful PyTorch implementations of both the proposed method and a set of baseline models, all evaluated under the open-world continual learning setting.

Datasets. We evaluate our method on two widely-used benchmark datasets: **Split CIFAR-100** [98] and **5-Datasets** [99], both of which are standard in the continual learning and open-world recognition literature.

- **Split CIFAR-100.** This dataset divides the original CIFAR-100 dataset into 10 distinct tasks, with each task comprising 10 specific classes. The original CIFAR-100 dataset contains a total of 60,000 samples, of which 50,000 samples are designated for the training set, and the remaining 10,000 samples are reserved for the test set. Consequently, each single task is allocated 5,000 training samples and 1,000 test samples. In the context of open detection, the test set of the current task is merged with the test set of the subsequent task to form the open test set for the current task. As a result, except for the final task, each task consists of 2,000 open test samples. Therefore, our setting allows for the evaluation and assessment of the model's performance on open classes during the training process.

- **5-Datasets.** We utilize a challenging dataset comprising TinyImagenet, NotMNIST, CUB200, Fashion-MNIST, and CIFAR10. Each dataset is treated as an individual task. Compared to Split CIFAR-100, all the datasets in the 5-Datasets exhibit wider variations, resulting in a greater significant forgetting issue. TinyImagenet consists of 200 classes with a total of 110,000 images, including 100,000 for training and 10,000 for testing. NotMNIST is an uncleaned dataset with 10 classes and comprises 18,256 training samples and 459 test samples. CUB200 is a widely used dataset for fine-grained visual classification tasks. It contains 11,788 images divided into 200 subcategories of birds, with 5,994 images used for training and 5,794 for testing. Fashion-MNIST includes 70,000 images from 10 categories: t-shirt, trousers, pullover,

dress, coat, sandal, shirt, sneaker, bag, and ankle boot. The dataset consists of 60,000 training samples and 10,000 test samples. CIFAR10 comprises RGB color images of 10 categories with 50,000 training images and 10,000 test images.

For open detection, we adopt the same setting as Split CIFAR-100, with a balanced number of known and unknown samples. Therefore, the number of open test sets for the top 4 tasks is as follows: 918, 918, 11,588, and 20,000.

Baselines. Since Open-world Continual Learning combines continual learning with novelty detection [25], we evaluate two key tasks: (1) *Unknown Detection* (identifying novel classes), and (2) *Known Classification* (learning sequentially without forgetting).

Unknown Detection. For detecting unknown classes, we compare several state-of-the-art methods. *OpenMax* [15] calibrates softmax probabilities to estimate openness, while *MSP* [80] uses the maximum softmax probability as an out-of-distribution (OOD) score. *MCD* [100] measures model disagreement on unlabeled data to quantify openness, and *EnergyBased* [63] uses energy scores to separate in-distribution (ID) and OOD data. *Entropy* [74] employs softmax entropy for uncertainty-based detection, whereas *MaxLogits* [101] improves OOD detection by relying on pre-softmax logits. Finally, *ODIN* [62] enhances MSP through temperature scaling and input perturbations to sharpen softmax boundaries.

Known Classification. For known-class classification, we benchmark against prominent continual learning approaches. Regularization-based methods include *EWC* [21], which penalizes changes to critical weights from past tasks, and *MAS* [96], which autonomously estimates parameter importance. Replay-based strategies like *DER++* [102] combine knowledge distillation with rehearsal, while *GEM* [44] enforces gradient constraints to prevent forgetting. Distillation-focused methods such as *LwF* [49] transfer knowledge across tasks via pseudo-rehearsal, and *HAT* [43] uses task-specific attention masks to isolate learned features. Bayesian approaches like *UCL* [46] incorporate variational inference for uncertainty-aware updates, and prompt-based techniques like *L2P* [32] and *DualPrompt* [83] use dynamic prompt tuning to adapt a frozen pre-trained backbone to sequential tasks.

Metrics. For unknown detection, we report the average area under the ROC curve (AUC_N), which gives us a single-number summary of how well the model separates known from unknown classes across all confidence thresholds, along with the average false positive rate (FPR_N) [74], where AUC_N measures open-class detection performance across all tasks and FPR_N specifically quantifies how often unknown samples are mistakenly accepted as known classes.

For known classification, we evaluate average final accuracy (A_N) across all learned classes and average forgetting rate (F_N) [32], with A_N reflecting overall retention

capability and F_N capturing the stability-plasticity trade-off by measuring performance degradation over sequential tasks.

4.3.1 Experimental Results and Analysis

Table 4.2: Evaluation results of unknown detection performance (%) across 10 sequential tasks (10 classes each) on Split CIFAR-100 and 5 tasks on 5-Datasets.

Dataset	Methods	AUC_N	FPR_N
Split CIFAR-100 (10 tasks)	OpenMax	49.99	50.06
	MSP	50.29	52.18
	MCD	50.17	70.37
	EnergyBased	49.66	71.86
	Entropy	49.56	51.17
	MaxLogits	52.63	74.99
	ODIN	50.81	52.64
	Pro-KT w/o task IDs	<u>91.01</u>	<u>41.31</u>
	Pro-KT	**92.69**	**39.71**
5-Datasets (5 tasks)	OpenMax	60.09	84.23
	MSP	51.22	87.96
	MCD	50.60	98.62
	EnergyBased	49.38	87.99
	Entropy	39.49	96.34
	MaxLogits	66.78	73.12
	ODIN	49.63	97.91
	Pro-KT w/o task IDs	<u>82.76</u>	<u>50.05</u>
	Pro-KT	**88.60**	**45.70**

Results on Unknown Detection (Q1). Table 4.2 presents the unknown detection performance across all evaluated methods. The results reveal two key findings: (1) existing baselines exhibit significantly degraded performance in the open-world continual learning setting, demonstrating their inability to maintain robust open-class detection capabilities across sequential tasks, and (2) our proposed *Pro-KT* achieves consistent and substantial improvements over all competitors, as evidenced by both the increased AUC_N and reduced FPR_N metrics across all experimental configurations.

Further analysis of threshold learning strategies highlights the importance of task-adaptive detection. When evaluating *Pro-KT* without task identifiers, we observe a notable performance decline in AUC_N, confirming that task-specific knowledge plays a crucial role in maintaining detection accuracy. This finding suggests that explicit task information helps refine decision boundaries for open-class detection in continual learning scenarios.

The superior performance of *Pro-KT* can be attributed to two main factors: (1) its effective knowledge transfer mechanism for open-class data across tasks, and

Table 4.3: Classification accuracy (%) on known classes for Split CIFAR-100 and 5-Datasets benchmarks. Experiments evaluate continual learning performance across 10 sequential tasks (10 classes each) on Split CIFAR-100, and 5 sequential tasks for 5 datasets. We used the ResNet32 and ViT backbones in the experiments.

Backbone	Method	Split CIFAR-100 (10 tasks)			5-Datasets (5 tasks)		
		$A_N(\uparrow)$	$Diff(\downarrow)$	$F_N(\downarrow)$	$A_N(\uparrow)$	$Diff(\downarrow)$	$F_N(\downarrow)$
ResNet32	*Upper-bound*	*72.99*	-	-	93.59	-	-
	EWC	34.16	38.83	36.20	51.25	42.34	42.06
	GEM	24.93	48.06	42.04	39.06	54.53	56.47
	LwF	16.73	56.26	51.62	36.18	57.41	59.19
	DER++	31.03	41.96	22.63	45.59	48.00	47.82
	HAT	30.43	42.56	36.73	37.00	56.59	51.35
	UCL	29.77	43.22	31.99	31.71	61.88	58.89
	MAS	29.58	43.41	22.14	49.59	44.00	33.90
ViT	*Upper-bound*	*86.07*	-	-	80.59	-	-
	LwF	71.93	14.14	<u>6.89</u>	19.02	61.57	44.23
	MAS	76.71	9.36	13.02	65.13	15.46	13.5
	L2P	80.63	5.44	10.93	69.06	11.59	<u>9.53</u>
	DualPrompt	<u>83.18</u>	<u>2.89</u>	8.42	<u>70.14</u>	<u>10.45</u>	10.11
	Pro-KT	**84.07**	**2.00**	**5.43**	**71.70**	**8.89**	**5.19**

(2) the innovative prompt bank design that progressively accumulates and organizes task-relevant knowledge. Notably, our method not only improves detection accuracy but also addresses the critical challenge of adaptive threshold selection through its dynamic strategy. These advantages collectively enable *Pro-KT* to set a new state-of-the-art for open-world continual learning.

Results on Known Samples Classification (Q1). We evaluate classification accuracy on known classes under continual learning settings. Notably, *Pro-KT* operates without requiring task identifiers during inference, making it suitable for task-agnostic class-incremental scenarios. For reference, we include an *Upper-bound* performance (offline joint training of all tasks) [32], regarded as the theoretical performance limit.

Table 4.3 presents comparative results on Split CIFAR-100 and 5-Datasets. Our analysis reveals three key findings: (1) *Pro-KT* consistently surpasses all baselines, establishing new state-of-the-art performance across both benchmarks; (2) When using ViT as backbone, *Pro-KT* achieves superior A_N compared to ResNet32, benefiting from both the strong representation capacity of large-scale pre-trained models and our method's effective prompt-based knowledge transfer mechanism; and (3) While baseline methods show non-trivial performance gaps relative to the upper-bound, *Pro-KT* demonstrates the smallest discrepancy, along with the lowest forgetting rate (F_N) across sequential tasks.

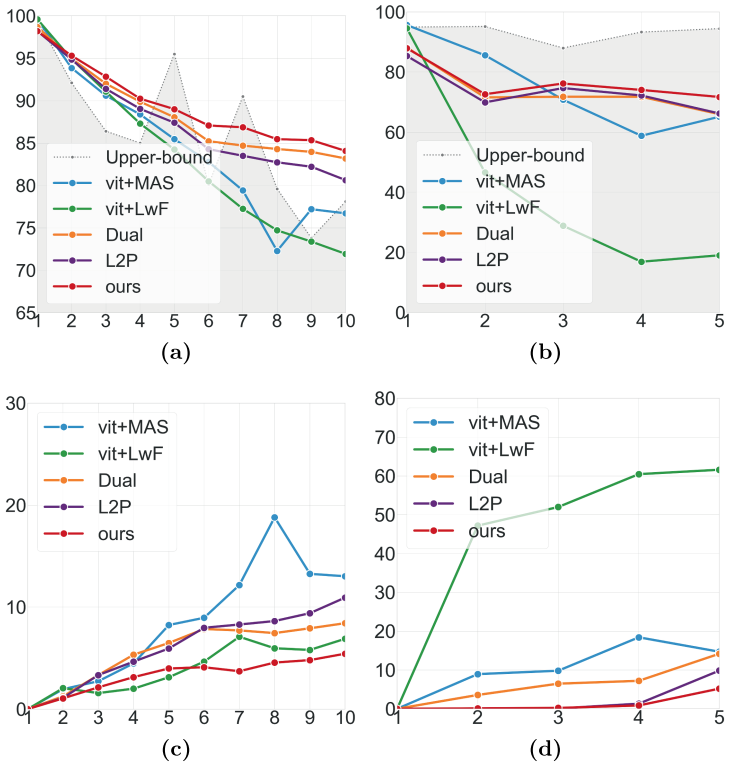


Figure 4.4: (a) and (b): Variation curve of A_N on Split CIFAR-100 and 5-Datasets, the x-axis indicates the task IDs. (c) and (d): Variation curve of F_N on Split CIFAR-100 and 5-Datasets, the x-axis indicates the task IDs.

4.3.2 Additional Results

Task Effect (Q1). Figure 4.4 presents the evolution of A_N and F_N metrics across sequential tasks. The results demonstrate *Pro-KT*'s consistent high performance across tasks and its superiority over baselines as the number of tasks increases. In fact, our method maintains consistently high accuracy (A_N) as new tasks are introduced, and it exhibits significantly lower forgetting (F_N) compared to baseline approaches. The performance gap widens progressively with increasing task complexity. These results highlight *Pro-KT*'s dual capability: effectively preventing catastrophic forgetting while sustaining high classification accuracy. The demonstrated performance superiority underscores the critical importance of our knowledge transfer mechanism in solving open-world continual learning challenges, particularly in scenarios with expanding task sequences.

Ablation Study (Q2). To demonstrate the importance of each key component in *Pro-KT*, we perform ablation studies focusing on three critical aspects: task identifiers for threshold determination, the complete task-aware open-set bound-

ary mechanism, and the prompt bank architecture. The experimental results are presented in Table 4.4.

First, we examine the impact of task identifiers by removing them during threshold determination. Compared to the full *Pro-KT* implementation, this results in a 1.8% drop in AUC_N on Split CIFAR-100 (from 92.69 to 91.01) and a more pronounced 6.6% drop on 5-Datasets (from 88.60 to 82.76). These results indicate that task identifiers contribute to more effective knowledge transfer by providing disambiguation across task boundaries, which becomes increasingly important when dealing with heterogeneous task distributions. Furthermore, we observe a slight increase in FPR_N under both settings, suggesting that removing task IDs can also exacerbate confusion in unknown detection.

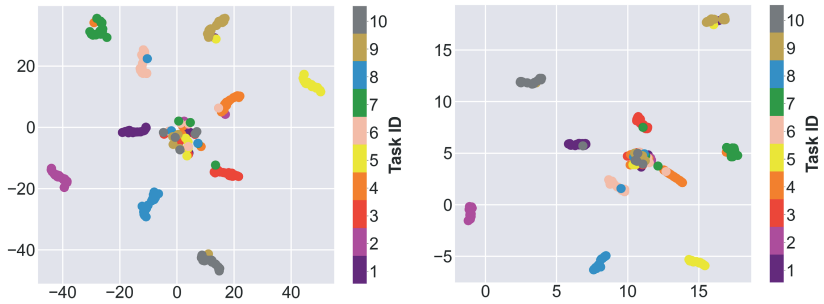
Second, we investigate the open-set boundary by completely removing this component. The consequences are severe: the model loses its open detection capability entirely, as shown by the AUC_N score dropping to approximately 50 (equivalent to random guessing) and the FPR_N reaching its highest observed value. This demonstrates the fundamental importance of the task-aware boundary mechanism for maintaining open-world detection performance.

Finally, we assess the prompt bank's contribution by replacing it with direct fine-tuning of the pre-trained backbone and task-specific classifiers. This ablation leads to a dramatic performance decline, revealing that naive fine-tuning approaches are fundamentally unsuited for open-world continual learning scenarios due to catastrophic forgetting. The significant performance gap underscores the prompt bank's crucial role in enabling effective, continual learning while preserving previously acquired knowledge.

Visualization (Q2). To investigate the knowledge transfer mechanism and characterize the different knowledge types learned by our prompt bank, we visualize all prompts using both t-SNE [103] and UMAP [104] dimensionality reduction techniques on Split CIFAR-100.

Table 4.4: Ablation study on the variants of *Pro-KT*.

Unknown	Split CIFAR-100		5-Datasets	
Detection	AUC_N	FPR_N	AUC_N	FPR_N
w/o task IDs	91.01	41.31	82.76	50.05
w/o AODB	50.02	87.63	46.78	88.90
Pro-KT	**92.69**	**39.71**	**88.60**	**45.70**
Known	Split CIFAR-100		5-Datasets	
Classification	A_N	F_N	A_N	F_N
w/o prompt bank	40.49	17.21	15.42	40.19
Pro-KT	**84.07**	**5.43**	**71.70**	**5.19**



(a) t-SNE **(b)** UMAP

Figure 4.5: Visualization of the prompt bank via T-SNE and UMAP after training on all tasks (Split CIFAR-100 dataset).

We analyze the prompt bank from the final model after sequential training on all tasks $[1, ..., n, ..., N]$. Figure 4.5 reveals two distinct patterns in the prompt embedding space. First, we observe clear clustering of prompts by task (indicated by color separation), demonstrating successful acquisition of task-specific knowledge. Second, a subset of prompts forms a central cluster containing representations learned across multiple tasks, providing direct evidence of task-agnostic knowledge capture.

These visualization results confirm that our prompt bank simultaneously maintains both specialized task knowledge and transferable general knowledge. The spatial organization of prompts in the embedding space validates the model's ability to preserve task-specific features while developing shared representations that facilitate effective knowledge transfer across the entire task sequence.

Parameter Sensitivity Analysis (Q3). Recall that there are three key parameters in our *Pro-KT*, namely the total number M of prompts learned from each task, the length L_p of a single prompt, and the selection size K used to prepend the input. Hence, $M \times N$ gives the total size of the prompt bank. Figure 4.6 shows the results on two datasets.

We now analyze the impact of different parameter settings that could affect the proposed *Pro-KT*. From the results in Figure 4.6 (left and middle), we observe that an oversized selection size K may lead to knowledge under-fitting. This phenomenon could be associated with prompt redundancy, where selecting too many prompts may introduce conflicting or irrelevant information, thereby diluting the useful task-relevant knowledge. While we find that the number of training steps L_P has little effect on performance, suggesting that prompt tuning itself is stable across moderate training durations.

Regarding the total number of prompts per task, we find that a reasonable capacity of the prompt bank is essential for encoding both task-common and task-specific

knowledge. From the results in Figure 4.6 (right), it is observed that in Split CIFAR-100, where the 10 tasks are similar, the performance remains stable across different values of M. However, in the 5-Datasets benchmark, where the tasks are highly dissimilar, increasing M introduces irrelevant prompts and exacerbates interference, resulting in degraded performance. We thus conclude that for tasks with greater dissimilarity, the hyperparameter M becomes more sensitive and requires careful tuning.

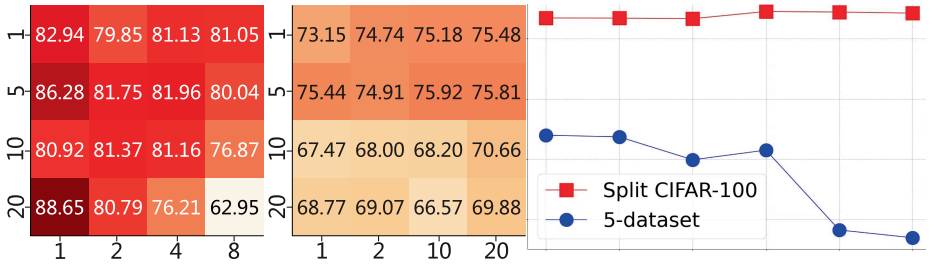


Figure 4.6: Left and middle are performance heatmaps (A_N) showing interaction effects between prompt length (L_P, y-axis) and selection size (K, x-axis) on both datasets, with fixed $M = 25$. Right is A_N variation across prompt counts (M) with optimal $L_P = 5$ and $K = 3$.

4.4 Summary

In this chapter, we introduced a prompt-based method named *Pro-KT*, designed to facilitate effective knowledge transfer through a structured prompt bank tailored for open-world continual learning. Our approach enables encoding of both task-specific and task-generic knowledge by associating learned prompts with a frozen pre-trained backbone. To support task-aware open-set decision-making, we further proposed two adaptive thresholding strategies that leverage prompt-derived representations. Extensive experiments across multiple benchmarks validate the effectiveness of *Pro-KT*, demonstrating robust performance in diverse task sequences and robustness under open-world conditions.

To better understand the limitations of our approach and guide future research directions, we conducted an in-depth analysis of representative failure cases. One notable limitation is the architectural dependence on prompt-compatible backbones. While *Pro-KT* performs strongly with transformer-based backbones (e.g., ViT), its AUC_N drops by 15.8% on Split CIFAR-100 and 21.9% on 5-Datasets when applied to ResNet-based backbones. This performance gap stems from the lack of pre-trained prompt compatibility in convolutional architectures, which impedes effective prompt activation and knowledge transfer. Addressing this challenge calls for more generalizable prompt integration strategies beyond transformers.

Another limitation arises from assumptions in the experimental setup. Specifically, we evaluated *Pro-KT* under class-balanced conditions, while real-world open-world continual learning tasks often exhibit severe imbalance, with unknown classes being rare or episodic. Our results (Section 4.4) show that under such conditions, the AUC_N of *Pro-KT* declines by up to 12.4%, and the false positive rate increases significantly, suggesting reduced model sensitivity to rare unknowns. Tackling this issue will require methods that can better exploit sparse unknown signals and incorporate fine-grained uncertainty modeling, especially in the low-resource regime.

In light of these findings, the next chapter explores a more challenging yet practically relevant setting—open-world continual learning under limited supervision. Specifically, we investigate how the core mechanisms of *Pro-KT* can be adapted to few-shot scenarios, where both known and unknown samples are scarce. This setting demands more expressive prompt representations, stronger generalization across tasks, and robust adaptation under data scarcity—all of which we address through principled model extensions and targeted empirical analysis.