Open-world continual learning via knowledge transfer

Li, Y.

Chapter 3

Preliminaries

In this chapter, we establish the foundational definitions and formal problem statements for Open-world Continual Learning (OWCL) that underpin our research. We also introduce key technical concepts that will be needed throughout this thesis.

To begin with, we present widely-adopted definitions of three core paradigms: Continual Learning (CL), Open-Set Learning (OSL), and Out-of-Distribution (OOD) detection. Building upon these, we then formalize the basic definitions and notation specific to open-world continual learning, which are used throughout all subsequent chapters. Additional chapter-specific definitions will be introduced in their respective contexts.

3.1 General Definitions

Before formalizing the core components of open-world continual learning, we first present the definitions of its two basic paradigms: continual learning and open-set recognition. These concepts, while often studied independently, are combined in a complex interaction in open-world continual learning systems.

3.1.1 Continual Learning

Continual Learning is formally defined as a learning paradigm where models adapt to evolving data distributions across sequential tasks. In this framework, the training samples arrive in temporal order, with each task t potentially drawn from a distinct distribution $\mathbb{D}_t$. A continual learning model $f_\theta(\cdot)$, parameterized by θ, must learn new tasks with minimal access to previous training data and maintain performance on all previously encountered tasks.

We adopt the following notation: tr and te denote training and testing phases, respectively. In addition, $\mathcal{D}_t^{tr} = \{\mathcal{X}_t^{tr}, \mathcal{Y}_t^{tr}\}$ represents the' training set of task t, where $\mathcal{X}_t^{tr}$ are the input samples, $\mathcal{Y}_t^{tr}$ their corresponding labels, and $t \in \{1, ..., T\}$

the task identifiers. Finally, $\mathbb{D}_t^{tr} := P_t^{tr}(\mathcal{X}, \mathcal{Y})$ denotes the true data distribution for task t.

Under the standard continual learning assumption, the testing distribution $\mathbb{D}_t^{te}$ matches $\mathbb{D}_t^{tr}$ for each task t. This leads to our formal definition:

Definition 1 ***Continual Learning.*** *Let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_t, \dots, \mathcal{T}_N\}$ be a sequence of learning tasks, where each task $\mathcal{T}_t$ comes with its own training data $\mathcal{D}_t^{tr}$ consisting of input-output pairs $(\boldsymbol{x}_i, y_i)$.*

A continual learning model learns a parameterized function $f_{\theta_t} : \mathcal{X} \to \mathcal{Y}$ by minimizing the expected loss:

$$\mathcal{L}(\theta_t) = \mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim \mathcal{D}_t^{tr}}[\ell(f_{\theta_t}(x_i), y_i)],$$

where ℓ is a task-specific loss (e.g., cross-entropy). To prevent forgetting knowledge from previous tasks $\mathcal{T}_1, \dots, \mathcal{T}_{t-1}$ learning is additionally constrained by

$$\mathcal{L}_t^{CL}(\theta) = \underbrace{\mathcal{L}_t(\theta)}_{\text{Current task}} + \lambda \sum_{k=1}^{t-1} \underbrace{\Omega_k(\theta, \theta_k^*)}_{\text{Memory of past tasks}} .$$

Here, Ω_k acts as a "memory anchor" [21], encoding knowledge from the task $\mathcal{T}_k$, and λ is a trade-off parameter that regulates the balance between plasticity (learning new tasks) and stability (retaining previous knowledge). Empirically, appropriate values of λ (e.g., in the range $[0.3, 0.7]$) have been shown to effectively mitigate catastrophic forgetting while preserving learning flexibility, as we further demonstrate in the experimental analysis presented in Chapters 4 to 7.

The final goal is to learn one single function f_{θ^*} that works well in all tasks:

$$\theta^* = \arg\min_{\theta} \sum_{t=1}^{N} \mathcal{L}_t^{CL}(\theta).$$

Continual learning aims to acquire new tasks sequentially without catastrophic forgetting of previously learned knowledge. To achieve this, typical "memory anchors" Ω_k are used, including regularization, replay (store old examples), or architectural (separate storage areas per task).

In real-world applications, two critical pieces of information may be unavailable during training: the task labels $\mathcal{Y}_{tr}^t$ and the task identities t. This practical constraint leads to three fundamental continual learning scenarios, categorized by data arrival patterns and task identifier availability [6]:

Task-Incremental Learning (TIL): In this setting, the model receives explicit task identifiers during both training and inference phases. This task-awareness enables the use of dedicated output modules for each task, typically implemented through a shared backbone network with multiple task-specific classifier heads. For

Table 3.1: Overview of the three CL scenarios.

Scenario	Required at test time
Task-IL	Solve tasks so far, task-ID provided
Domain-IL	Solve tasks so far, task-ID not provided
Class-IL	Solve tasks so far *and* infer task-ID

instance, a visual recognition system might maintain separate output layers for different domains (e.g., one head for medical images, another for satellite images) while sharing convolutional feature extractors.

Domain-Incremental Learning (DIL): In this scenario, task identity is not provided at test time, so that the model only needs to perform the correct task without identifying which one it is. The assumption is that the task structure remains fixed across domains, while the input distributions vary. A practical example would be an agent operating across multiple environments, where the task/environment identification is not required, while data distributions change as they depend on the specific environment.

Class-Incremental Learning (CIL): As the most demanding scenario, class-incremental learning requires models to not only solve all previously seen tasks but also infer which task a given input belongs to. The system must gradually incorporate new categories while preserving performance on previously learned ones, without access to task identifiers. This mirrors real-world applications like retail product recognition, where new items are continuously added to inventory while maintaining accurate identification of existing products.

3.1.2 Open-Set Learning and Out-of-Distribution Detection

Traditional supervised learning operates under a closed-world assumption, where the label spaces for training ($\mathcal{Y}_{\text{train}}$) and testing are identical. Formally, a model learns a mapping $f : \mathcal{X} \to \mathcal{Y}_{\text{train}}$ with the guarantee that all test samples belong to $\mathcal{Y}_{\text{train}}$. While effective for constrained environments, this assumption fails in realistic scenarios where systems encounter new categories during deployment.

To address this limitation, various strategies have been proposed in recent years. Two prominent and increasingly studied directions are **Open-Set Learning** that extends classification to include rejection of unknown categories, and **Out-of-Distribution Detection** that focuses on distinguishing in-distribution from out-of-distribution samples. Though both approaches handle distributional shifts, they differ fundamentally in their objectives and formulations.

Definition 2 ***Open-Set Learning.*** *In open-set learning, the test-time label space is extended to include unknown classes:*

$$\mathcal{Y}_{test} = \mathcal{Y}_{train} \cup \mathcal{Y}_{unknown}, \quad \mathcal{Y}_{unknown} \cap \mathcal{Y}_{train} = \emptyset.$$

The system must simultaneously achieve two objectives: (1) accurate classification of samples $x \in \mathcal{X}$ from a known class $x \in \mathcal{Y}_{train}$), and (2) reliable detection or rejection of samples from unknown classes $y \in \mathcal{Y}_{unknown}$.

The above dual requirement implies that an open-set learning model not only maintains discriminative power for established categories but also identifies novel inputs that deviate from the training distribution. OOD detection instead creates binary in/out decision boundaries without explicit class differentiation.

Definition 3 ***Out-of-Distribution Detection.*** *Let $\mathcal{D}_{in}$ be the in-distribution data (training distribution), and $\mathcal{D}_{out}$ be any distribution such that $supp(\mathcal{D}_{out}) \cap supp(\mathcal{D}_{in}) = \emptyset$. The goal of OOD detection is to learn a function $g : \mathcal{X} \to \{0, 1\}$ such that:*

$$g(x) = \begin{cases} 1 & \textit{if } x \sim \mathcal{D}_{in} \ (\textit{in-distribution}) \\ 0 & \textit{if } x \sim \mathcal{D}_{out} \ (\textit{out-of-distribution}) \end{cases}$$

Optionally, if x is classified as in-distribution, it may be passed to a classifier f for label prediction.

Following the two definitions above, Table 3.2 contrasts open-set learning and out-of-distribution detection across several dimensions. Although both paradigms address the limitations of closed-set assumptions, they diverge fundamentally in their objectives and implementations.

Open-set learning methods like OpenMax [15, 57] and C2AE [78] focus on discriminative classification of known classes while rejecting unknowns within an expanded label space. In contrast, out-of-distribution detection techniques such as ODIN [62], Mahalanobis distance [79], and MSP [80] establish binary decision boundaries to separate in-distribution from out-of-distribution samples, without explicit class differentiation.

These foundational concepts share a common structural framework, where data is partitioned into base known classes available during training and unknown classes encountered only during deployment, with no overlap between them by definition. This difference between known and unknown categories forms the base of our investigation into open-world continual learning.

Our work extends these principles by developing novel knowledge transfer mechanisms that bridge continual learning and open-set recognition. At its core, the framework addresses three interconnected challenges: the preservation of base-class knowledge during incremental updates, the transfer of discriminative features between known and unknown categories, and the maintenance of an optimal stability-plasticity balance as the class space evolves dynamically. These objectives are pursued while respecting the inherent constraints of real-world deployment scenarios.

The **open-world environment** we consider significantly generalizes conventional open-set learning and out-of-distribution settings through its temporal dimension,

Table 3.2: Comparison between OSL and OOD Detection.

Aspect	**OSL**	**OOD**
Objective	Classify known classes and reject unknowns	Detect whether a sample belongs to the training distribution
Label Space at Test	$\mathcal{Y}_{\text{train}} \cup \mathcal{Y}_{\text{unknown}}$	No specific label set for out-of-distribution samples; only binary in/out decision
Output	Class label or rejection	Binary label: in-distribution vs. out-of-distribution
Assumption	Unknown classes exist and are not seen during training	Outliers come from a distribution disjoint from training data
Application Focus	Open-world classification, e.g., novel class discovery	Model safety, anomaly detection, robust deployment
Representative Methods	OpenMax [57], C2AE [78]	ODIN [62], MSP [80], Mahalanobis [79]

where both base and unknown classes may evolve across learning episodes, and its emphasis on the dynamic interplay between familiar and novel categories. This perspective, which formalizes the interdependence of incremental learning and open-set recognition, is formally defined in the next section.

3.2 Problem Definition

To formally characterize the open-world continual learning challenge, Definition 4 gives the learning objective as a double optimization problem. This formulation fundamentally extends traditional continual learning in two critical aspects: first, by explicitly accounting for open-class samples during testing through an open-set risk term; and second, by requiring simultaneous minimization of both this detection risk and the standard incremental learning error across all tasks. The resulting framework captures the essential tension between maintaining existing knowledge while remaining responsive to novel, previously unseen categories.

Definition 4 ***(OWCL Problem Formulation.)*** *Consider a task t characterized by a training set $\mathcal{D}_t^{tr} = \{\boldsymbol{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}_t\}_{i=1}^{N_t}$ with M_t classes, and a test set $\mathcal{D}_t^{te} = \{\boldsymbol{x}_j \in \mathcal{X}, y_j \in \mathcal{Y}_t'\}_{j=1}^{N_t'}$ where $\mathcal{Y}_t' \supset \mathcal{Y}_t$ (i.e., it contains additional unknown classes). During training, an open-world continual learning model only accesses $\mathcal{D}_t^{tr}$, while during testing it must handle both known ($\mathcal{Y}_t$) and unknown ($\mathcal{Y}_t' \backslash \mathcal{Y}_t$) classes. Let $h(\cdot)$ be a feature extractor mapping inputs to a latent space, and define $\mu(h(\boldsymbol{x}), h(\mathcal{D}_t^{tr}))$ to be the open-set risk measuring detection capability for unknowns, and $\epsilon_{\mathcal{D}_t^{tr}}(h)$ to be the prediction error on known classes. The open-world continual learning objective* *learns an optimal $h^* \in \mathcal{H}$ that minimizes these two factors across all tasks:*

$$h^* = \arg\min_{h \in \mathcal{H}} \{(1-\lambda) \sum_{t=1}^{T} \underbrace{\mu(h(\boldsymbol{x}), h(\mathcal{D}_t^{tr}))}_{\textit{Open-set risk}} + \lambda \sum_{t=1}^{T} \underbrace{\epsilon_{\mathcal{D}_t^{tr}}(h)}_{\textit{Incremental error}} \} \tag{3.1}$$

where $\mathcal{H}$ is the hypothesis space of possible models, $\lambda \in [0, 1]$ is a trade-off parameter between old and new knowledge, $\epsilon_{\mathcal{D}_t^{tr}}(h)$ is the generally suggested prediction error term on task t [15], μ can be defined on any scoring function for out-of-distribution detection, and T is the total number of tasks.

In practice, μ can be instantiated using a variety of out-of-distribution detection scores, such as energy-based scores, Mahalanobis distance, or margin-based confidence [63, 79]. The choice of μ depends on the specific nature of the feature space and the distribution shift between known and unknown classes.

As formalized in Definition 4, the open-world continual learning paradigm considers a (potentially infinite) sequence of tasks, each associated with a training set containing only known-class examples. The model is required to incrementally learn from these tasks while maintaining the ability to recognize both previously seen and unseen (novel) classes at test time. Therefore, a promising open-world continual learning model must not only accurately classify samples from known classes, but also reliably detect inputs belonging to previously unseen (unknown) classes.

To do this, the model learns a feature extractor that maps inputs into a useful representation space. The learning objective then tries to find the best such feature extractor by balancing two goals: keeping the prediction error on known classes low (called incremental error), and being good at recognizing unknown or out-of-distribution inputs (called open-set risk). A parameter λ adjusts how much the model focuses on remembering what it has learned versus being cautious about unfamiliar data, and this balancing act happens continually as new tasks arrive.

To fix our formal problem formulation and facilitate discussion of the proposed methods, Table 3.3 provides a summary of the key mathematical notations used throughout this thesis.

3.3 Four OWCL Scenarios

Traditional continual learning is typically divided into three scenarios based on whether task identifiers are available at test time: *TIL*, *DIL*, and *CIL*.

However, as outlined earlier in Chapter 1, in the context of open-world continual learning, such assumptions no longer hold. During testing, the model cannot rely on task identifiers, as unknown or open-category samples may appear arbitrarily alongside known ones. Moreover, data distributions in open-world continual learning evolve dynamically through both novel class emergence and recurring open samples.

Table 3.3: Notations used in the OWCL problem formulation.

Symbol	Description
t	Index of the current task
T	Total number of tasks encountered so far
θ_k or (θ_k^*)	Parameters optimized after learning task t
$\mathcal{L}_t(\theta)$	Expected loss on task t
$\Omega_t(\theta, \theta_t)$	Constraint/regularization term to retain knowledge from task t
$\mathcal{D}_t^{tr}, \mathcal{D}_t^{te}$	Training, test set for task t
N_t	Number of training samples for task t
N_t'	Number of test samples for task t
M_t	Number of classes in the training set of task t
M_t'	Number of classes in the test set of task t
$\mathcal{Y}_{i=1}^{M_t}$	Label set for training samples in task t
$\mathcal{Y}_{i=1}^{M_t'}$	Label set for test samples in task t
$\boldsymbol{x}$	A sample input from the input space $\mathcal{X}$
$h_\theta(\cdot)$ or $f_\theta(\cdot)$	Latent feature extractor or encoder function
h^*	The optimal function to be learned
$\mu(\cdot, \cdot)$	Open risk function, e.g., based on out-of-distribution scoring
$\epsilon_{\mathcal{D}_t^{tr}}(h)$	Prediction error on task t's training set
$\ell(\cdot, \cdot)$	Task-specific loss function (e.g., cross-entropy)
$\mathcal{H}$	Hypothesis space (function space)
λ	Trade-off hyperparameter balancing open risk and prediction error

This renders conventional continual learning scenarios inadequate for open-world continual learning settings.

Hence, we formulate four progressively challenging open-world continual learning scenarios to reflect different patterns of task evolution and unknown-class recurrence in open-world environments:

- **CINO: Class-Incremental Learning with Non-Repetitive Open Samples**. In CINO, each task introduces a disjoint set of new training classes, and the open-category samples encountered during testing do not reappear in future tasks. The model learns each class only once, without access to past data or recurring unknowns. This scenario is particularly suited for one-pass learning applications where data cannot be revisited, such as species identification in biodiversity monitoring or new material classification in industrial inspection.

- **CIRO: Class-Incremental Learning with Repetitive Open Samples.** In CIRO, training classes remain disjoint across tasks as in CINO, but open-category samples that appear during testing can recur in future tasks. This setting requires the model to incrementally accumulate knowledge about recurring unknowns without revisiting known class labels. CIRO is representative of real-world settings such as autonomous driving or robotic exploration, where unknown objects (e.g., road signs, dynamic obstacles) may resurface over time, and adaptive open-set handling is essential.

- **KINO: Knowledge-Incremental Learning with Non-Repetitive Open Samples.** KINO relaxes the disjoint-class assumption, allowing training classes to repeat across tasks, and includes distributional shifts over time. However, open-category samples remain non-repetitive—they appear only once during testing and do not recur. This scenario models data streams where user preferences or content characteristics evolve, such as in personalized recommendation systems or adaptive spam filtering, where the model must cope with shifting distributions while handling unseen content once.

- **KIRO: Knowledge-Incremental Learning with Repetitive Open Samples.** KIRO represents the most complex and realistic open-world continual learning scenario, where both known-class distributions and open-category samples can change and recur over time. *While KINO handles shifting distributions of known classes, KIRO additionally requires models to recognize and consolidate knowledge about recurring unknowns.* This setting reflects the demands of long-term, real-world systems such as intelligent surveillance, financial fraud detection, or user behavior modeling, where evolving patterns—whether known or unknown—must be continually tracked, recognized, and adapted to.

Remarks. Compared to task-, domain-, and class-incremental learning settings, knowledge-incremental learning introduces significantly greater complexity. It requires the model to learn new classes incrementally without access to task identifiers, while also maintaining robustness to shifts in data distributions across tasks. Moreover, the unpredictable presence of open-category data means the model must not only recognize unfamiliar classes but also transfer knowledge across both known and unknown examples. These combined challenges make Open World Continual Learning fundamentally more demanding than traditional continual learning paradigms.

3.4 Pretrained Models in OWCL: A Brief Overview

As outlined above, this thesis aims to push continual learning into open-world scenarios, where data distributions shift over time and the model must continually handle previously unseen instances. Within this context, we explore how knowledge can be effectively transferred under such open and dynamic conditions.

The rise of large-scale pretrained models (PTMs) has brought a paradigm shift to both the continual learning and open-set recognition communities, including in tasks such as out-of-distribution detection. These models are increasingly used as foundational backbones, followed by task-specific fine-tuning. This shift has underscored the critical influence of the backbone architecture on the learning behavior of open-world continual learning models. Furthermore, several open-world continual learning methods are tightly coupled to specific backbone designs, which limits their generalizability across architectures.

To establish the foundations for the open-world continual learning framework proposed in this thesis, this section provides a concise overview of recent PTM-based methods in continual learning. This review both contextualizes our approach within the current research landscape and motivates the architectural choices made in the design of our proposed method.

3.4.1 Attention Mechanism and Transformer Architecture

In recent years, pretrained models have shown strong performance in continual learning, attracting significant attention from the research community. However, these approaches still suffer from catastrophic forgetting, primarily because importance weights associated with previous tasks cannot be recovered once overwritten—even when using a pretrained backbone. To mitigate this issue, many studies have explored Parameter-Efficient Fine-Tuning (PEFT) techniques, which enable model adaptation to new tasks through minimal parameter updates, thereby preserving the knowledge stored in the frozen backbone.

In the vision domain, current methods commonly employ a pretrained Vision Transformer (ViT) [81] as a fixed feature extractor f_θ. The ViT architecture is built upon a sequence of multi-head self-attention layers [82], where attention scores are computed via a scaled dot-product mechanism, allowing the model to capture complex relationships across image patches.

Definition 5 ***Scaled Dot-Product Attention.*** *Let $K \in \mathbb{R}^{N \times d_k}$ be a key matrix with N key vectors, and $V \in \mathbb{R}^{N \times d_v}$ be a value matrix with N corresponding value vectors. Given a query matrix $Q \in \mathbb{R}^{M \times d_k}$, attention over (K, V) is defined as*

$$Attention(Q, K, V) = softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{3.2}$$

where the softmax function acts on the rows of matrix $QK^\top \in \mathbb{R}^{M \times N}$.

The Scaled Dot-Product Attention mechanism forms the core computational unit of transformer architectures, enabling models to weigh the relevance of different input elements when producing contextualized representations. However, relying on a single attention function may limit the model's capacity to capture diverse relationships. To address this, transformers, including (ViTs), use Multi-head Self-Attention (MSA), which extends the basic attention operation by computing multiple attention

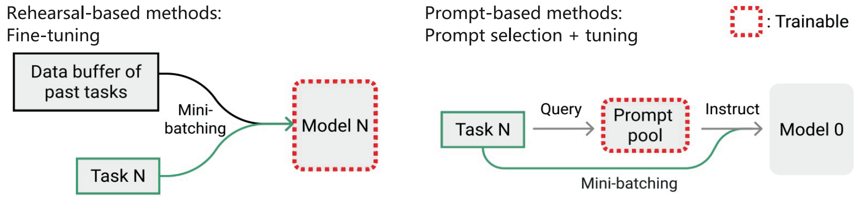

Figure 3.1: In contrast to rehearsal-based continual learning approaches that sequentially adapt full or partial model parameters to new tasks and rely on a rehearsal buffer to alleviate catastrophic forgetting, prompt-based methods adopt a unified backbone model augmented by learned prompts [32, 83]. Task-specific information is encapsulated within the prompts, thereby eliminating the necessity of a rehearsal buffer. Furthermore, different methods have different ways of selecting and updating prompts dynamically, usually enabling task-agnostic inference at test time without explicit task identity.

functions in parallel. Each head attends to different subspaces of the input representation, allowing the model to capture richer and more nuanced dependencies. The formal definition of the MSA layer is given below.

Definition 6 ***Multi-head Self-Attention (MSA) Layer.*** *Let X^Q, X^K, X^V denote the input query, key, and value matrix, respectively, where $X^Q = X^K = X^V = [x_1, ..., x_N]^\top \in \mathbb{R}^{N\times d}$, and N is the length of the input sequence. The output is expressed as*

$$MSA(X^Q, X^K, X^V) := Concat(h_1, ..., h_m)W^O \in \mathbb{R}^{N\times d}, \tag{3.3}$$

$$h_i := Attention(X^Q W_i^Q, X^K W_i^K, X^V W_i^V), \quad i \in [m] \tag{3.4}$$

where $W^O \in \mathbb{R}^{md_v \times d}$, $W_i^Q \in \mathbb{R}^{d\times d_k}$, $W_i^K \in \mathbb{R}^{d\times d_k}$, and $W_i^V \in \mathbb{R}^{d\times d_v}$ are projection matrices, and m is the number of heads in the MSA layer. In ViTs, they use $d_k = d_v = d/m$.

3.4.2 Prompt-based CL Methods with PEFT Techniques

Building on the transformer-based backbone discussed above, prompt-based tuning has emerged as a flexible, lightweight, and rehearsal-free approach for adapting frozen models to new tasks in continual learning settings [32, 83, 84]. Instead of modifying the entire network, these methods introduce a small number of trainable parameters, known as prompts, that augment the model's input representations. These prompts serve to encode task-specific knowledge and interact with the attention computations in the PTM-based backbone, enabling efficient adaptation while preserving previously learned capabilities. This design enables task-specific knowledge adaptation efficiently, making it particularly suitable for continual learning with limited memory or parameter sharing.

As illustrated in Figure 3.1, prompt-based approaches [83, 84] enhance model flexibility by integrating learnable prompt vectors into the Multi-Head Self-Attention mechanism. These prompt tokens are injected into the query, key, and value matrices at each attention layer, effectively conditioning the model on new tasks. Let the prompt parameters be denoted by $p \in \mathbb{R}^{L_p \times d}$, where L_p is the prompt length and d denotes the hidden dimensionality.

According to prior work [83], two main formulations have been widely adopted: Prompt Tuning (ProT) [85] and Prefix Tuning (PreT) [86]. In Prompt Tuning, the same prompt vectors are concatenated to the inputs of the query, key, and value projections simultaneously. In contrast, Prefix Tuning splits the prompt into two components, $p^K, p^V \in \mathbb{R}^{\frac{L_p}{2} \times d}$, which are prepended to the key and value matrices only, leaving the query stream unchanged. This decoupled design enables finer control over attention flow and has demonstrated improved performance across various downstream tasks:

$$\begin{aligned} f_{\text{prompt}}^{\text{Pre-T}}(p, \mathbf{X}^Q, \mathbf{X}^K, \mathbf{X}^V) &:= \text{MSA}\left(\mathbf{X}^Q, \begin{bmatrix} p^K \\ \mathbf{X}^K \end{bmatrix}, \begin{bmatrix} p^V \\ \mathbf{X}^V \end{bmatrix}\right) \\ &= \text{Concat}(\tilde{h}_1, ..., \tilde{h}_m)W^O. \end{aligned} \tag{3.5}$$

A broad spectrum of prompt-based continual learning methods has been developed to mitigate catastrophic forgetting by dynamically introducing new prompts as learning progresses. In this paradigm, prompts act as task-specific adapters: for each new task, a set of adaptive prompts is either generated or selected to guide the model's behavior. This modular design enables the model to retain task-relevant knowledge and reuse it effectively when encountering related inputs. At inference time, appropriate prompt configurations can be retrieved to enable accurate prediction on previously learned tasks [32].

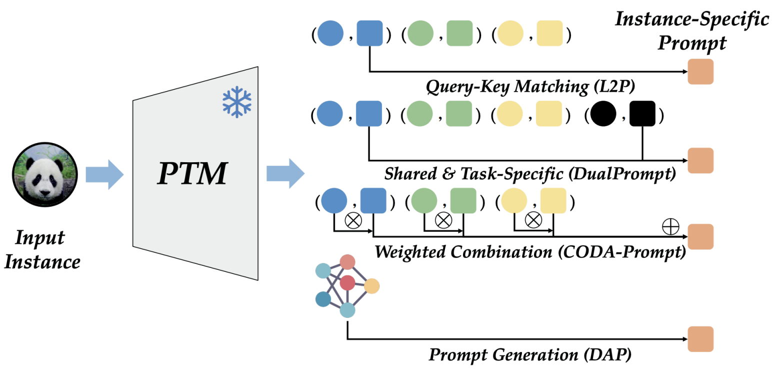

Figure 3.2: Different kinds of prompt selection, including key-value matching, shared and task-specific retrieval, attention-based combination, and instance-specific prompt generation.

An overview of prompt-based continual learning methods is provided in [87], and visually summarized in Figure 3.2. This figure categorizes a range of prompt selection strategies, including hard prompt retrieval (L2P) [32], structured prompt grouping (DualPrompt) [83], attention-based soft aggregation (CODA Prompt) [88], and meta-generated prompts (DAP) [89].

One of the earliest contributions in this line of work, L2P [32], incorporates prompt learning into the continual learning setting by introducing a shared pool of learnable prompts. A key-query matching mechanism dynamically selects a subset of prompts from the pool based on the current input, enabling input-dependent adaptation. Building upon this, DualPrompt [83] introduces a structured decomposition of the prompt space into General Prompts (G-Prompts), which encode task-agnostic knowledge, and Expert Prompts (E-Prompts), which specialize in capturing task-specific features. This separation encourages generalization while preserving task-level specificity.

Similarly, S-Prompt [90] adopts a task-specific prompt learning strategy, maintaining a separate prompt for each task, akin to L2P but without retrieval. In contrast, CODA-Prompt [88] enhances flexibility by growing the prompt pool over time and using an attention-based mechanism to softly aggregate prompts. Each prompt is weighted by task-specific attention scores, allowing the model to form context-aware combinations instead of relying on hard selection.

Moving beyond explicit retrieval mechanisms, DAP [89] introduces a meta-network, typically a multilayer perceptron (MLP), that generates prompts conditioned on input representations. This instance-level prompt generation enables fine-grained adaptation and removes the need for a fixed prompt pool.

Recent studies have also explored alternative strategies, such as appending all available prompts to the input or adopting *visual prompts*, i.e., low-level pixel-space modifications to the image itself. Furthermore, the rise of multimodal pre-trained models has led to approaches that leverage textual information to guide prompt selection or generation, particularly within vision-language pretraining frameworks.

A notable advancement in this space is HiDe-Prompt [84], which redefines the continual learning objective through a hierarchical decomposition. By optimizing each sub-objective independently, HiDe-Prompt enhances both performance and modularity, establishing a new benchmark for continual image classification.

Therefore, inspired by the aforementioned prompt-based continual learning frameworks, we further investigate their applicability and extensibility under the open-world continual learning setting. In open-world continual learning, models are expected to make inferences without access to task identifiers and to distinguish between known and unknown classes. Prompt-based methods naturally align with the demands of open-world continual learning, providing a task-agnostic solution to this challenge. For instance, CODA-Prompt's soft selection strategy enables inference without relying on explicit task labels, aligning well with open-world continual learning's open-set and label-free nature. Similarly, dynamic prompt generation methods like DAP offer instance-level adaptability, which is essential when encountering unknown or evolving data distributions. These properties position prompt-based continual learning methods as promising candidates for scalable, open-world scenarios, particularly in settings with data privacy constraints where revisiting past samples is infeasible and task-agnostic.

3.4.3 Other PTMs-based CL Methods with PEFT Techniques

Recent advances in parameter-efficient continual learning studies extend beyond prompt-based methods, with adapter-based approaches demonstrating particular success. Works like SimpleCIL [91], ADAM [92], and EASE [93] employ lightweight modular architectures [94] that attach task-specific adapter layers to frozen pretrained backbones. These adapters achieve dual objectives: encoding new task knowledge through localized parameter updates, while preserving existing representations via regularization that anchors new features near previously learned embeddings. This dual mechanism effectively balances plasticity and stability, minimizing interference during sequential learning.

Complementing these methods, vision-language models like CLIP [95] provide inherent open-world capabilities through their contrastive image-text pretraining. Their zero-shot classification ability, enabled by measuring alignment between visual features and textual prompts (e.g., "a photo of a dog"), offers natural open-set detection through confidence thresholding. However, CLIP's static architecture faces two key limitations in open-world continual learning settings: (i) fixed decision boundaries that cannot adapt to evolving task distributions, and (ii) domain gaps when applied to specialized datasets. Our proposed Pro-KT and MOB frameworks address these gaps through dynamic prompt tuning and adaptive hypersphere boundaries, respectively.

These developments reflect a broader paradigm shift toward modular, efficient adaptation of foundation models. Contemporary approaches increasingly favor lightweight task-specific components (prompts/adapters) over full model retraining, explicit mechanisms for knowledge preservation during incremental updates, and unified architectures that support both closed-set and open-set recognition.

Our work contributes to this direction by introducing preference-conditioned adaptation techniques into the PTM backbone. Specifically, our method dynamically adjusts to task requirements through learned gating mechanisms, maintains backward compatibility with previously learned representations, and jointly optimizes for both known class accuracy and unknown detection. This approach demonstrates superior adaptability, robustness, and generalization capabilities compared to static architectures, as evidenced by our experiments simulating dynamic, unpredictable, and open-ended data environments.