UNLOCKING PATCH-LEVEL FEATURES FOR CLIP-BASED CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Class-Incremental Learning (CIL) enables models to continuously integrate new knowledge while mitigating catastrophic forgetting. Driven by the remarkable generalization of CLIP, leveraging pre-trained vision-language models has become a dominant paradigm in CIL. However, current work primarily focuses on aligning global image embeddings (*i.e.*, [CLS] token) with their corresponding text prompts (*i.e.*, [EOS] token). Despite their good performance, we find that they discard the rich patch-level semantic information inherent in CLIP's encoders. For instance, when recognizing a *rabbit*, local patches may encode its distinctive cues, such as long ears and a fluffy tail, which can provide complementary evidence for recognition. Based on the above observation, we propose SPA (**S**emantic-guided **P**atch-level **A**lignment) for CLIP-based CIL, which aims to awaken long-neglected local representations within CLIP. Specifically, for each class, we first construct representative and diverse visual samples and feed them to GPT-5 as visual guidance to generate class-wise semantic descriptions. These descriptions are used to guide the selection of discriminative patch-level visual features. Building upon these selected patches, we further employ optimal transport to align selected patch tokens with semantic tokens from class-wise descriptions, yielding a structured cross-modal alignment that improves recognition. Furthermore, we introduce task-specific projectors for effective adaptation to downstream incremental tasks, and sample pseudo-features from stored class-wise Gaussian statistics to calibrate old-class representations, thereby mitigating catastrophic forgetting. Extensive experiments demonstrate that SPA achieves state-of-the-art performance.

1 INTRODUCTION

In recent years, the rapid development of deep learning (He et al., 2016; Liu et al., 2015) has profoundly impacted numerous fields. However, real-world scenarios are inherently dynamic, with data often appearing as continuous and non-stationary streams (Rebuffi et al., 2017; Zhao et al., 2020). In such contexts, traditional deep learning models can suffer from catastrophic forgetting (French, 1999; Serra et al., 2018; Shi et al., 2021), as they tend to overwrite previously learned knowledge when trained on shifting data distributions. To address this issue, Class-Incremental Learning (CIL) (Gao et al., 2022; Zhou et al., 2024b; 2025a) has emerged as an essential paradigm that enables models to incrementally absorb new concepts while retaining previously acquired knowledge. Recently, the rapid advancement of pre-trained models (Dosovitskiy et al., 2020; Li et al., 2023), particularly Vision-Language Models (VLMs) (Yao et al., 2022), like CLIP (Radford et al., 2021), has significantly accelerated a paradigm shift in the CIL (Wang et al., 2022d; Zhou et al., 2024a). By harnessing their powerful generalization capabilities, current research (Huang et al., 2025; Zhou et al., 2025c) has shifted from training networks from scratch to applying Parameter-Efficient Fine-Tuning (PEFT) strategies (Hu et al., 2025) on frozen PTM backbone networks, requiring the adjustment of only a small number of additional parameters (Jia et al., 2021).

CLIP (Radford et al., 2021) leverages a contrastive learning paradigm to project images and texts into a shared embedding space. Its powerful zero-shot generalization and representation learning capabilities allow it to effectively handle various downstream tasks. Specifically, its image encoder prepends a learnable [CLS] token to the input sequence of image patches to aggregate global visual features, while the text encoder relies on the [EOS] token to capture global textual semantics. Although both encoders generate rich patch-token and word-token sequences during the encoding

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

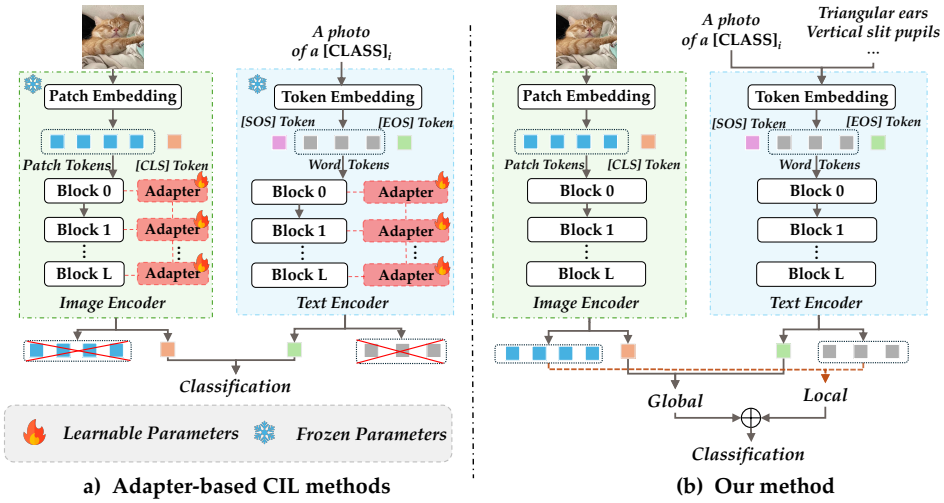


Figure 1: Comparison of CLIP-based CIL paradigms. (a) Existing adapter-based CIL methods mainly rely on the global [CLS] and [EOS] tokens for cross-modal alignment, leaving the rich semantic information in patch-level tokens insufficiently explored. (b) Our method explicitly aligns patch-level tokens, activating the previously overlooked local features and improving performance.

process (Wu et al., 2024a), the CLIP model only computes the cosine similarity between the global [CLS] and [EOS] embeddings for classification. Current CLIP-based CIL research (Huang et al., 2025) primarily focuses on PEFT strategies (*e.g.*, via adapters or prompt tuning) (Gao et al., 2024; Wu et al., 2024b), as illustrated in Fig. 1(a). While these methods have improved CIL performance, they also inherently inherit the global alignment paradigm of the original CLIP model, leaving rich local visual features and semantic contexts underutilized. For instance, when recognizing a *rabbit*, different local patches may encode distinctive visual cues such as long ears, a short fluffy tail, or soft fur texture. Explicitly leveraging these patch-tokens can provide additional evidence for recognition.

Awakening and utilizing the local information carried by token-level image-text offers a promising approach to enhance the discriminative capabilities in CIL, but it also presents two significant challenges: **1)** The original image patch-tokens are often filled with a considerable amount of background noise (Chen et al., 2022), and simply extracting all the patch information without proper filtering can interfere with classification tasks by introducing irrelevant features. This makes it difficult for the model to effectively leverage the rich spatial details embedded within the patches, leading to suboptimal performance. **2)** Directly aligning local patches with global text prompts (*e.g.*, "a photo of a [CLASS]") suffers from a semantic level mismatch. The class prompt encodes global semantics, whereas patch features correspond to local regions. Forcing these local visual representations to match the same global semantics ignores their semantic diversity, making different patches less distinguishable and weakening the effectiveness of patch-level features in classification.

To address the above challenges, we propose SPA (**S**emantic-guided **P**atch-level **A**lignment) for CLIP-based CIL. Different from existing methods, SPA explicitly activates the long-neglected patch-level representations within CLIP and leverages them to improve performance in CIL. Specifically, we first construct representative and diverse visual samples for each class, and then leverage GPT-5 to generate class-wise attribute semantics from these samples. Based on these semantics, SPA evaluates the semantic relevance of image patches and selects the top-K most discriminative patch-level visual features. We then align these selected patch-level features with multiple class-wise semantic embeddings via optimal transport, thereby achieving structured cross-modal alignment. To mitigate catastrophic forgetting, we further introduce expandable task-specific projectors and Gaussian pseudo-feature sampling. The projectors support task-wise adaptation by updating only newly added modules, while the calibration module samples pseudo-features from stored class-wise statistics to preserve old-class distributions. Experimental results show that SPA significantly improves the discriminative power in CIL, surpassing existing methods across nine benchmark datasets.

2 Related Work

Class-Incremental Learning. CIL aims to enable models to continuously learn new classes from a data stream while mitigating catastrophic forgetting (De Lange et al., 2021; Masana et al., 2022).

Traditional CIL methods can be broadly categorized into several groups. Regularization-based methods (Aljundi et al., 2018; Kirkpatrick et al., 2017; Zenke et al., 2017) impose constraints on key updated parameters, penalizing changes in the critical parameter space. Replay-based methods mitigate forgetting by preserving a memory buffer (Chaudhry et al., 2018; Luo et al., 2023) of exemplar samples from previous tasks or by employing generative models (Ostapenko et al., 2019; Xiang et al., 2019) to reconstruct and replay prior data distributions. Dynamic network-based methods expand the network structure to accommodate new tasks, including neuron expansion (Xu & Zhu, 2018; Yoon et al., 2017), backbone expansion (Wang et al., 2022a; Zheng et al., 2025), and prompt expansion (Liu et al., 2021). Knowledge distillation-based methods (Hinton et al., 2015; Li & Hoiem, 2017) transfer knowledge from previously trained models to the current one, ensuring that new tasks are learned while retaining important information from earlier tasks.

Pre-Trained Model-Based CIL. With the rapid development of PTMs, such as ViT (Dosovitskiy et al., 2020) and CLIP (Radford et al., 2021), the research focus of CIL has gradually shifted from conventional training with random initialization to an efficient adaptation paradigm based on PTMs (Li et al., 2025; Qi et al., 2025). To adapt to downstream tasks while preserving the generalization capability, existing ViT-based methods typically freeze the pre-trained backbone and introduce lightweight learnable modules (Smith et al., 2023; Wang et al., 2022b), such as learnable prompts prepended to the input token sequence (Wang et al., 2022d) or adapters inserted into selected layers of the Transformer backbone (Fukuda et al., 2025; Yu et al., 2024). CLIP-based methods (Zhou et al., 2025c) further exploit the cross-modal alignment capability of vision-language models to improve CIL performance. RAPF (Huang et al., 2024) adaptively calibrates old-class representations and alleviates catastrophic forgetting through decomposed parameter fusion after adapter tuning. CLG-CBM (Yu et al., 2025) builds a language-guided concept bottleneck model, leveraging interpretable concept representations to improve CIL performance while enhancing model interpretability.

3 PRELIMINARIES

Class-Incremental Learning. Formally, CIL aims to endow models with the ability to continuously acquire new knowledge from B distinct incremental tasks, denoted as $\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^B\}$, while retaining previously learned information (Rebuffi et al., 2017). At the b-th task, the model is trained on a dataset $\mathcal{D}^b = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_b}$ consisting of n_b examples, where $\mathbf{x}_i \in \mathbb{R}^D$ represents an instance of class $y_i \in Y_b$, with Y_b denoting the label space corresponding to task b. The label spaces of different tasks are disjoint (*i.e.*, $Y_b \cap Y_{b'} = \varnothing$ for $b \neq b'$). In this paper, we adopt the exemplar-free protocol (Wang et al., 2022c; Zhou et al., 2022), where at each task b, the model has access only to the training data of the current task $\mathcal{D}^b$, and cannot access any data from previous tasks $\mathcal{D}^{1:b-1}$. The goal of CIL is to build a unified classifier for all seen classes $\mathcal{Y}_b = Y_1 \cup \cdots \cup Y_b$. Formally, we aim to learn a model $f(\mathbf{x}) : X \to \mathcal{Y}_b$ that minimizes the expected risk:

$$f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}}\, \mathbb{E}_{(\mathbf{x},y)\sim \mathcal{D}_t^1 \cup \cdots \mathcal{D}_t^b} \mathbb{I}\left(y \neq f(\mathbf{x})\right) , \tag{1}$$

where $\mathcal{H}$ represents the hypothesis space, $\mathbb{I}(\cdot)$ denotes the indicator function, and $\mathcal{D}_t^b$ refers to the data distribution of the b-th task.

Vision-Language Model. Following (Huang et al., 2024; Zhou et al., 2025c), we use the pre-trained model CLIP (Radford et al., 2021) as the initialization for $f(\mathbf{x})$. CLIP consists of an image encoder $g_i(\cdot)$ and a text encoder $g_t(\cdot)$, which project images and texts into a shared d-dimensional embedding space. Given an input image $\mathbf{x} \in \mathbb{R}^D$, the image encoder $g_i(\cdot)$ (*e.g.*, Vision Transformer (Dosovitskiy et al., 2020)) first divides it into M non-overlapping image patches, which are linearly projected with positional encodings. A learnable class token (*i.e.*, [CLS] token) is then prepended to the image embeddings, forming the input sequence for the Transformer layers. Through multi-head self-attention, the [CLS] token aggregates global information from the patch embeddings and serve as the global visual representation. Meanwhile, each patch embedding interacts with other patch-tokens to enrich its own representation, ultimately yielding the feature embedding sequence $\mathbf{V} = [\mathbf{v}_{\text{cls}}, \mathbf{v}_1, \cdots, \mathbf{v}_M]$, where $\mathbf{v}_{\text{cls}} = g_i(\mathbf{x}) \in \mathbb{R}^d$ is the global visual representation, and $\{\mathbf{v}_i\}_{i=1}^M$ corresponds to the patch-tokens containing rich local information that remains underutilized. Similarly, for a given class $y_i \in \mathcal{Y}_b$, we construct a templated prompt $\mathbf{c}_i$ (*e.g.*, "a photo of a [CLASS]"). The text encoder $g_t(\cdot)$ tokenizes the prompt, prepends a start token (*i.e.*, [SOS] token) and appends an end token (*i.e.*, [EOS] token), encoding it into the sequence of embeddings $\mathbf{T}^i = [\mathbf{t}^i_{\text{sos}}, \mathbf{t}^i_1, \cdots, \mathbf{t}^i_N, \mathbf{t}^i_{\text{eos}}]$, where $\mathbf{t}^i_{\text{eos}} = g_t(\mathbf{c}_i) \in \mathbb{R}^d$ serves as the global semantic representation. During the inference phase, CLIP relies on computing the similarity between the global visual

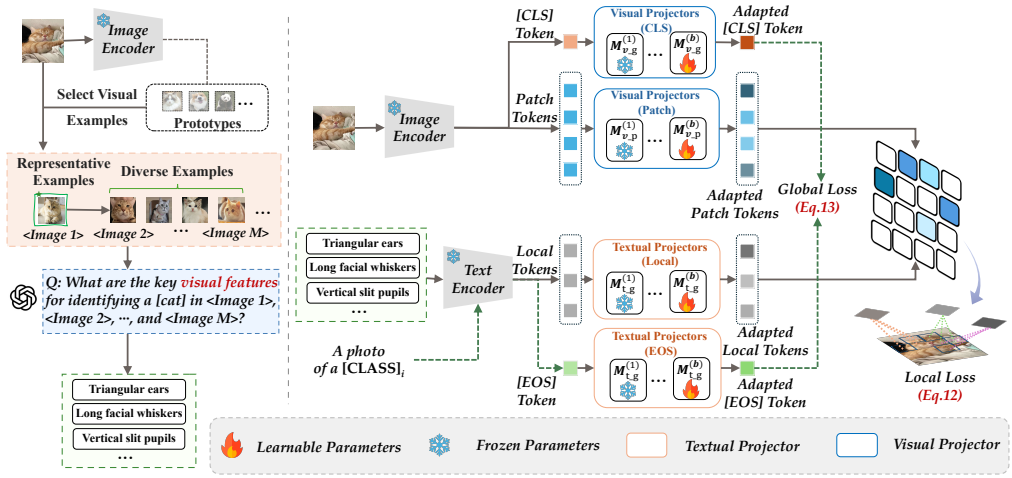


Figure 2: Illustration of SPA. **Left:** For each class, SPA first constructs representative and diverse visual samples and uses GPT-5 to generate class-wise semantics. **Right:** These semantics are then used to select discriminative image patches, and SPA performs structured local alignment via optimal transport with task-specific projectors to mitigate catastrophic forgetting.

feature $\mathbf{v}_{\text{cls}}$ and the global text embeddings $\mathbf{t}_{\text{eos}}$ of each class in $\mathcal{Y}_b$ for category prediction. The probability that the image $\mathbf{x}$ belongs to class y_i is calculated as:

$$f_{y_i}(\mathbf{x}, \mathbf{c}_i) = \frac{\exp\left(\cos\left(\mathbf{v}_{\text{cls}}, \mathbf{t}^i_{\text{eos}}\right)/\tau\right)}{\sum_{j=1}^{|\mathcal{Y}_b|}\exp\left(\cos\left(\mathbf{v}_{\text{cls}}, \mathbf{t}^j_{\text{eos}}\right)/\tau\right)} = \frac{\exp\left(\cos\left(g_i(\mathbf{x}), g_t(\mathbf{c}_i)\right)/\tau\right)}{\sum_{j=1}^{|\mathcal{Y}_b|}\exp\left(\cos\left(g_i(\mathbf{x}), g_t(\mathbf{c}_j)\right)/\tau\right)}, \tag{2}$$

where $\cos(\cdot,\cdot)$ represents the cosine similarity and τ denotes the temperature parameter. In Eq. 2, cross-modal alignment relies solely on [CLS] token $\mathbf{v}_{\text{cls}}$ and [EOS] token $\mathbf{t}_{\text{eos}}$, while the rich local visual patch-token features $\{\mathbf{v}_i\}_{i=1}^M$ and textual token-level semantics $\{\mathbf{t}^i_j\}_{j=1}^N$ are ignored.

Baselines in Class-Incremental Learning. Existing CIL methods using pre-trained models primarily rely on PEFT strategies to adapt to new tasks while mitigating catastrophic forgetting. Representative approaches include prompt-based methods (Wang et al., 2022b;c;d), which introduce a learnable prompt pool $\mathcal{P}$, and adapter-based methods (Fukuda et al., 2025; Gao et al., 2024; Tan et al., 2024), which insert lightweight trainable adapters $\mathbf{M}_v$ into the frozen Transformer backbone. Both paradigms refine the global representations (*e.g.*, $\tilde{\mathbf{v}}_{\text{cls}}$) using only a small number of learnable parameters, enabling efficient adaptation without updating the backbone.

Discussions. Although these paradigms achieve efficient adaptation to downstream tasks and mitigate catastrophic forgetting, they primarily inherit CLIP's global alignment mechanism in Eq. 2, leaving the richly informative patch-level features and token-level textual semantics underexplored. Therefore, explicitly exploiting these local features is crucial for enhancing recognition.

4 SPA: Semantic-guided Patch-level Alignment

Motivated by the underexplored local visual and textual semantics in CLIP-based CIL, we propose SPA, a semantic-guided patch-level alignment framework. As shown in Fig. 2, SPA first selects representative and diverse samples as visual references, and leverages GPT-5 to generate class-wise attribute semantics. Guided by these semantics, SPA performs discriminative patch selection and employs optimal transport to achieve structured alignment between selected visual tokens and textual attribute tokens. To mitigate forgetting, we further introduce task-specific visual and textual projectors, together with Gaussian pseudo-feature sampling for preserving old-class distributions.

4.1 Class-wise Attribute Semantic Generation

Most prior CLIP-based CIL methods construct text prompts solely from class names (*e.g.*, "A photo of a [CLASS]$_i$"). These prompts are encoded into global text representations for classification following Eq. 2. However, such prompts cannot fully capture visual attributes such as color and local structure, which are crucial for distinguishing categories. To provide richer textual priors, we construct class-wise attribute semantics for discriminative patch selection and local alignment.

Visual Sample Construction. To generate more discriminative class-wise attribute semantics, we

first construct a visual sample set for each class that is both representative and diverse. Specifically, for each class $c \in Y_b$ at task b, we first compute its visual prototype $\mathbf{p}_c$ based on the global features extracted by the frozen visual encoder $\bar{g}_i(\cdot)$:

$$\mathbf{p}_c = \frac{1}{N} \sum_{j=1}^{|\mathcal{D}^b|} \mathbb{I}(y_j = c)\, \bar{g}_i(\mathbf{x}_j), \tag{3}$$

where $N = \sum_{j=1}^{|\mathcal{D}^b|} \mathbb{I}(y_j = c)$ denotes the number of samples belonging to class c, $\mathbb{I}(\cdot)$ denotes the indicator function. We define the cosine distance between two features $\mathbf{a}$ and $\mathbf{b}$ as $d(\mathbf{a}, \mathbf{b}) = 1 - \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}$, where $\|\cdot\|_2$ denotes the L2-norm. Based on this prototype, we select the sample whose global feature is closest to the prototype $\mathbf{p}_c$ as the representative sample $\mathbf{x}_c^r$:

$$\mathbf{x}_c^r = \arg \min_{\mathbf{x}_j \in \mathcal{D}^b,\, y_j = c} d\left(\bar{g}_i\left(\mathbf{x}_j\right), \mathbf{p}_c\right), \tag{4}$$

However, relying on a single representative sample is insufficient to fully capture intra-class visual variations, and may introduce instance-specific bias into subsequent semantic construction. To enrich the diversity of the class-level visual context, we further select n samples that are most dissimilar to $\mathbf{x}_c^r$ from the remaining samples of the same class. The final visual sample set is constructed as:

$$\mathcal{S}_c = \{\mathbf{x}_c^r\} \cup \mathrm{Topn}_{\mathbf{x}_j \in \mathcal{D}^b, y_j = c,\, \mathbf{x}_j \neq \mathbf{x}_c^r} d\left(\bar{g}_i\left(\mathbf{x}_j\right), \bar{g}_i\left(\mathbf{x}_c^r\right)\right), \tag{5}$$

where $\mathrm{Topn}(\cdot)$ selects the top-n samples with the largest distances. In this way, $\mathbf{x}_c^r$ provides representative visual information, while the additional n samples capture the intra-class diversity. They jointly constitute the class-level visual context used for subsequent semantic generation.

Attribute Semantic Generation. Using the constructed class-level visual samples, we utilize GPT-5.4 (Singh et al., 2025) to generate attribute descriptions for each class. Specifically, for class c, its sample images and class name are fed into GPT-5.4 with the following prompt template to generate attribute semantics:

> **Input:** *Image 1, Image 2, $\cdots$ Image M, and [CLASS]$_c$*
> **Q***: What are the key visual features for identifying a [CLASS]$_c$ in these images? Focus on the most discriminative attributes.*
> **A***:* **1.** *Pointed triangular ears.* **2.** *Vertical-slit pupils.* **3.**$\cdots$

Based on the above prompt, we generate a set of discriminative visual attributes for class c, denoted as $\mathrm{A}_c = \{a_j^c\}_{j=1}^{N_a}$, where a_j^c denotes the j-th attribute of class c.

4.2 Semantic-guided Discriminative Patch Selection

The constructed attribute semantics provide class-specific textual priors for local representation learning. However, directly aligning them with all patch tokens may introduce background regions and class-irrelevant local patterns. For example, when recognizing a *rabbit* on grass, background grass patches may be incorrectly treated as rabbit-related regions, which introduce noise and interfere with local representation learning. To reduce such interference, we adopt semantic-guided discriminative patch selection to filter irrelevant patches and preserve informative local patches.

Discriminative Patch-level Set Selection. Given an input image $\mathbf{x}$, the visual encoder $\bar{g}_i(\cdot)$ extracts $[\mathbf{v}_{\text{cls}}, \mathbf{v}_1, \cdots, \mathbf{v}_M]$, from which we form the patch feature set $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^M$. For each class c, we use the generated attribute descriptions A_c as class-wise semantic guidance. We randomly sample N descriptions from A_c, and encode them with the text encoder $\bar{g}_t(\cdot)$ to obtain attribute embeddings $\mathcal{T}_c = \{\mathbf{t}_n^c\}_{n=1}^N$. We then compute the similarity between each patch-level feature and each attribute embedding of class c as:

$$\mathrm{sim}_{m,n}^c = \frac{\mathbf{v}_m^\top \mathbf{t}_n^c}{\|\mathbf{v}_m\|_2 \|\mathbf{t}_n^c\|_2}, \tag{6}$$

where $\mathrm{sim}_{m,n}^c$ measures the semantic relevance between the m-th patch in image $\mathbf{x}$ and the n-th attribute of class c. Based on these similarities, we further aggregate the responses of each image patch to all attribute semantics of class c, and define the class-aware discriminative score as $\mathrm{q}_m^c = \frac{1}{N} \sum_{n=1}^N \mathrm{sim}_{m,n}^c$. According to q_m^c, we select the top-K patches from image $\mathbf{x}$ that are most relevant to the attribute semantics of class c, and denote the discriminative patch set from image $\mathbf{x}$ as:

$$\mathcal{R}_c = \left\{ \mathbf{v}_m \,\middle|\, \mathrm{q}_m^c \in \mathrm{TopK}\left(\{\mathrm{q}_m^c\}_{m=1}^M\right) \right\}, \tag{7}$$

where $\mathrm{TopK}(\cdot)$ denotes the operation of selecting the K highest discriminative scores among all M patches. Through the above selection process, the model is encouraged to preserve class-relevant local features and reduce interference from background and irrelevant regions.

Task-specific Projector Construction. Although CLIP exhibits strong zero-shot generalization, the domain gap between pre-training data and downstream incremental tasks makes adaptation necessary. Therefore, we introduce lightweight projectors after the frozen visual and textual encoders. However, a single shared projector across all tasks may limit adaptation to new classes and increase interference with previously learned knowledge. To address this issue, we introduce task-specific projectors to enhance task-wise adaptation while alleviating interference with previously acquired knowledge. Specifically, for the b-th task, we initialize a new task-specific visual projector $\mathbf{M}_v^b(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ and a new task-specific textual projector $\mathbf{M}_t^b(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ after the frozen visual and textual encoders, respectively, while keeping all projectors $\{\mathbf{M}_v^{1:b-1}, \mathbf{M}_t^{1:b-1}\}$ learned from previous tasks frozen. Given a patch feature $\mathbf{v}_k$ and an attribute embedding $\mathbf{t}_n^c$, we obtain the adapted visual representation $\tilde{\mathbf{v}}_k$ and textual representation $\tilde{\mathbf{t}}_n^c$ by cumulatively aggregating the outputs of the current and historical projectors:

$$\tilde{\mathbf{v}}_k = \mathbf{M}_v^b(\mathbf{v}_k) + \sum_{i=1}^{b-1} \bar{\mathbf{M}}_v^i(\mathbf{v}_k), \quad \tilde{\mathbf{t}}_n^c = \mathbf{M}_t^b(\mathbf{t}_n^c) + \sum_{i=1}^{b-1} \bar{\mathbf{M}}_t^i(\mathbf{t}_n^c), \tag{8}$$

where $\bar{\mathbf{M}}_v^i(\cdot)$ and $\bar{\mathbf{M}}_t^i(\cdot)$ denote the frozen visual and textual projectors learned from task i, respectively. In this way, the current task can be adapted by newly introduced projectors, while the knowledge acquired from previous tasks is preserved through the frozen historical projectors. The aggregated features are then used to replace the raw visual and textual features in Eq. 6, and are further fed into the discriminative patch-level set selection and local alignment.

4.3 Optimal Transport-based Patch-level Alignment

After obtaining the discriminative patch set $\mathcal{R}_c$ in Sec. 4.2, directly matching the selected patches to textual attribute embeddings remains suboptimal. Naive matching strategies (Lafon et al., 2024) tend to force multiple distinct patches to concentrate on the same attribute semantics, leading to degenerate local alignment and feature redundancy. This weakens the stability of representation learning and aggravates forgetting of previously acquired knowledge. Therefore, it is essential to establish a balanced local alignment for robust representation learning in CIL.

Alignment via Optimal Transport. To address this issue, we formulate patch-level alignment as a joint assignment problem. By imposing marginal constraints, optimal transport (Gangbo & McCann, 1996) distributes the matching mass across patches and attributes, leading to more balanced local correspondences. Specifically, for class c, we rewrite the selected discriminative patch set from the input image as an ordered set $\mathcal{R}_c = \{\mathbf{r}_k^c\}_{k=1}^K$, where $\mathbf{r}_k^c \in \mathbb{R}^d$ denotes the feature of the k-th selected patch. The corresponding attribute embedding set is denoted as $\mathcal{T}_c = \{\mathbf{t}_n^c\}_{n=1}^N$. We first compute the pairwise semantic similarity $\mathrm{sim}_{k,n}$ between each selected patch $\mathbf{r}_k^c$ and each attribute embedding $\mathbf{t}_n^c$ using Eq. 6. Based on this similarity matrix, the transport cost matrix $\mathbf{C}_c \in \mathbb{R}^{K\times N}$ is defined as:

$$\mathbf{C}_c(k,n) = 1 - \mathrm{sim}_{k,n}, \tag{9}$$

To enforce balanced matching, we adopt uniform marginal distributions over the selected patches and attribute embeddings, i.e., $\mathbf{a} = \frac{1}{K}\mathbf{1}_K, \mathbf{b} = \frac{1}{N}\mathbf{1}_N$. Therefore, the entropy-regularized balanced optimal transport problem is formulated and solved via the Sinkhorn iterations (Cuturi, 2013):

$$\mathbf{\Pi}_c^* = \arg\min_{\mathbf{\Pi}} \langle \mathbf{\Pi}, \mathbf{C}_c \rangle - \lambda \mathcal{H}(\mathbf{\Pi}) \quad \text{s.t.} \quad \mathbf{\Pi}\mathbf{1}_N = \mathbf{a},\ \mathbf{\Pi}^{\mathrm{T}}\mathbf{1}_K = \mathbf{b}, \tag{10}$$

where $\mathcal{H}(\cdot)$ denotes the entropy regularization term, and $\lambda > 0$ is a hyperparameter. After obtaining the optimal transport matrix $\mathbf{\Pi}_c^*$, we define the local alignment score for class c as the transport-weighted semantic similarity between the selected patches and the attribute embeddings:

$$\sigma_c^{\mathrm{loc}} = \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbf{\Pi}_c^*(k,n)\,\mathrm{sim}_{k,n}. \tag{11}$$

We regard these scores of all seen classes as the local classification logits and apply softmax normalization to obtain $f_{\mathrm{local},c}(\mathbf{x})$. The local loss is defined as:

$$\mathcal{L}_l = \ell(f_{\mathrm{local}}(\mathbf{x}), y), \quad \text{where} \quad f_{\mathrm{local},c}(\mathbf{x}) = \frac{\exp(\sigma_c^{\mathrm{loc}})}{\sum_{c'=1}^{C}\exp(\sigma_{c'}^{\mathrm{loc}})}. \tag{12}$$

Table 1: Comparison of the average and last performance of different methods. The best results are highlighted in bold. Detailed complete results are reported in the Appendix.

Method	Aircraft				CIFAR100				Cars			
	B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
ZS-CLIP (Radford et al., 2021)	26.66	17.22	21.70	17.22	81.81	71.38	76.49	71.38	82.60	76.37	78.32	76.37
SimpleCIL (Zhou et al., 2025a)	59.24	48.09	53.05	48.09	84.15	76.63	80.20	76.63	92.04	86.85	88.96	86.85
L2P (Wang et al., 2022d)	47.19	28.29	44.07	32.13	82.74	73.03	81.14	73.61	76.63	61.82	76.37	65.64
DualPrompt (Wang et al., 2022c)	44.30	25.83	46.07	33.57	81.63	72.44	80.12	72.57	76.26	62.94	76.88	67.55
CODA-Prompt (Smith et al., 2023)	45.98	27.69	45.14	32.28	82.43	73.43	78.69	71.58	80.21	66.47	75.06	64.19
RAPF (Huang et al., 2024)	50.38	23.61	40.47	25.44	86.14	78.04	82.17	77.93	82.89	62.85	75.87	63.19
CLG-CBM (Yu et al., 2025)	66.05	55.93	59.25	55.39	86.58	80.15	83.59	79.28	93.25	88.76	90.11	88.19
PROOF (Zhou et al., 2025c)	63.81	56.14	59.47	57.10	86.77	79.11	83.32	79.73	90.74	86.51	88.00	85.58
BOFA (Li et al., 2026)	70.96	60.43	66.09	61.36	86.07	79.19	83.02	79.44	94.21	90.20	92.13	90.50
SPA (Ours)	**71.57**	**61.51**	**66.82**	**63.01**	**88.53**	**81.81**	**85.01**	**81.60**	**94.43**	**90.91**	**92.33**	**91.43**

Method	ImageNet-R				CUB				UCF			
	B0 Inc20		B100 Inc20		B0 Inc20		B100 Inc20		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
ZS-CLIP (Radford et al., 2021)	83.37	77.17	79.57	77.17	74.38	63.06	67.96	63.06	75.50	67.64	71.44	67.64
SimpleCIL (Zhou et al., 2025a)	81.06	74.48	76.84	74.48	83.81	77.52	79.75	77.52	90.44	85.68	88.12	85.68
L2P (Wang et al., 2022d)	75.97	66.52	72.82	66.77	70.87	57.93	75.64	66.12	86.34	76.43	83.95	76.62
DualPrompt (Wang et al., 2022c)	76.21	66.65	73.22	67.58	69.89	57.46	74.40	64.84	85.21	75.82	84.31	76.35
CODA-Prompt (Smith et al., 2023)	77.69	68.95	73.71	68.05	73.12	62.98	73.95	62.21	87.76	80.14	83.04	75.03
RAPF (Huang et al., 2024)	81.26	70.48	76.10	70.23	79.09	62.77	72.82	62.93	92.28	80.33	90.31	81.55
CLG-CBM (Yu et al., 2025)	84.64	78.50	81.46	77.88	85.37	78.24	77.74	76.97	95.04	91.36	94.17	91.85
PROOF (Zhou et al., 2025c)	83.84	78.40	81.20	78.92	82.31	76.64	79.20	76.37	94.58	91.10	93.58	90.91
BOFA (Li et al., 2026)	84.53	78.77	81.60	79.12	86.66	80.58	83.18	80.79	93.19	88.71	92.60	89.43
SPA (Ours)	**85.63**	**79.08**	**82.50**	**79.50**	**87.17**	**81.93**	**84.43**	**82.23**	**95.63**	**92.38**	**95.43**	**93.48**

Global Loss. To preserve stable global cross-modal alignment during incremental learning, we extract the global [CLS] token and the global [EOS] token from the frozen CLIP encoders, and transform them with a separate set of task-specific projectors for global semantic alignment, yielding the adapted global visual representation $\tilde{\mathbf{v}}_{\text{cls}}$ and textual representation $\tilde{\mathbf{t}}^c_{\text{eos}}$ using Eq. 8. Although task-specific projectors mitigate parameter interference across incremental tasks, the model may still be biased toward new classes. To mitigate this bias, we adopt Gaussian feature calibration (Zhu et al., 2021). For class c, we store only its class-wise global statistics, *i.e.*, the visual prototype $\mathbf{p}_c$ and covariance matrix $\mathbf{\Sigma}_c$. When learning task b, we sample old-class pseudo features from $\mathcal{N}(\mathbf{p}_c, \mathbf{\Sigma}_c)$, and combine them with current-task samples from $\mathcal{D}^b$ to optimize the global classification loss:

$$\mathcal{L}_g = \ell(f_{\text{global}}(\mathbf{x}), y), \quad \text{where} \quad f_{\text{global},c}(\mathbf{x}) = \frac{\exp(\cos(\tilde{\mathbf{v}}_{\text{cls}}, \tilde{\mathbf{t}}^c_{\text{eos}})/\tau)}{\sum_{j=1}^{C} \exp(\cos(\tilde{\mathbf{v}}_{\text{cls}}, \tilde{\mathbf{t}}^j_{\text{eos}})/\tau)}. \tag{13}$$

We apply Gaussian sampling only to global features, as patch-level features are instance-dependent and their patch-attribute correspondences are difficult to preserve under Gaussian sampling.

Summary. In SPA, we exploit the local representations in CLIP through semantic-guided selection and structured alignment. Class-wise semantics guide discriminative patch selection, while optimal transport establishes correspondences between these patches and textual attributes. To adapt to incremental tasks and mitigate forgetting, task-specific projectors are introduced for visual and textual features. During training, we jointly optimize the global loss and the local loss:

$$\mathcal{L} = \mathcal{L}_g + \beta \mathcal{L}_l, \tag{14}$$

where β is a hyperparameter balancing the two loss terms.

Inference. During inference, we aggregate the global and local logits to obtain the final prediction:

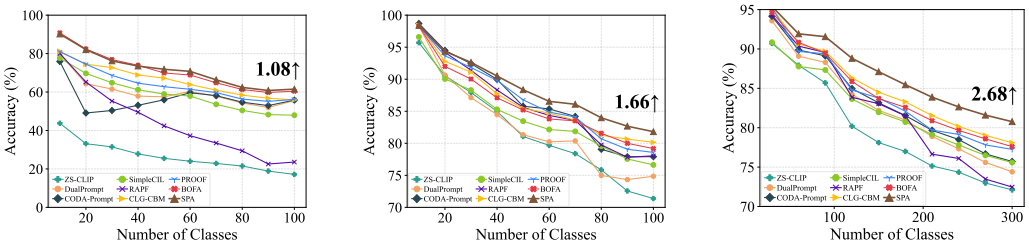
$$f(\mathbf{x}) = f_{\text{global}}(\mathbf{x}) + \beta f_{\text{local}}(\mathbf{x}). \tag{15}$$

5 Experiments

In this section, we evaluate SPA on nine benchmark datasets and compare it with state-of-the-art CIL methods. We then conduct ablation studies and additional analyses to examine the contribution of each component and the reliability of SPA. More experimental results are provided in the appendix.

5.1 Implementation Details

Dataset. Following (Zhou et al., 2025c; 2022), we evaluate our method on nine benchmark datasets, *i.e.*, CIFAR100 (Krizhevsky et al., 2009), FGVCAircraft (Maji et al., 2013), CUB200 (Wah et al., 2011), ObjectNet (Barbu et al., 2019), Food101 (Bossard et al., 2014), ImageNet-R (Hendrycks



(a) Aircraft Base0 Inc10 (b) CIFAR100 Base0 Inc10 (c) SUN Base0 Inc30

Figure 3: Incremental performance of different methods on the B0 setting. We report the performance gap after the last incremental stage of SPA and the runner-up method at the end of the line.

et al., 2021), StanfordCars (Krause et al., 2013), UCF101 (Soomro et al., 2012) and SUN397 (Xiao et al., 2010). Adopting the sampling strategy from (Zhou et al., 2025b;c), we select 100 classes from CIFAR100, Food101, FGVCAircraft, StanfordCars, and UCF101; 200 classes from ObjectNet, CUB200, and ImageNet-R; and 300 classes from SUN397.

Dataset split. We construct the CIL tasks under the widely adopted 'B-m Inc-n' protocol (Rebuffi et al., 2017; Wang et al., 2022d), where m and n denote the number of classes in the initial session and each subsequent incremental session, respectively. For fair comparison, the class order is randomly shuffled with a fixed random seed of 1993 and kept consistent across all evaluated baselines.

Comparison methods. To thoroughly evaluate the effectiveness of our proposed method, we compare it with SOTA CIL baselines. Specifically, these baselines include ViT-based methods, *e.g.* SimpleCIL (Zhou et al., 2025a), L2P (Wang et al., 2022d), DualPrompt (Wang et al., 2022c), and CODA-Prompt (Smith et al., 2023), as well as CLIP-based methods, *e.g.* RAPF (Huang et al., 2024), CLG-CBM (Yu et al., 2025), PROOF (Zhou et al., 2025c), and BOFA (Li et al., 2026). In addition, we use ZS-CLIP (Radford et al., 2021) as a performance reference for CLIP on downstream tasks.

Training details. All experiments are implemented in PyTorch (Paszke et al., 2019) using the C3Box (Sun & Zhou, 2026) toolbox. For fair comparison, **all methods use the same pre-trained CLIP backbone**, *i.e.*, ViT-B/16. For visual-only methods (*e.g.*, L2P, DualPrompt, CODA-Prompt) that cannot exploit textual priors, we initialize them with the visual branch of the same pre-trained CLIP backbone. We present the results based on LAION-400M pre-trained CLIP (Ilharco et al., 2021) in the main paper, while the corresponding results based on OpenAI pre-trained CLIP (Radford et al., 2021) are provided in the appendix. We train our method using SGD for 10 epochs with a batch size of 64 and an initial learning rate of 0.05, which is decayed using a cosine annealing schedule. By default, the number of selected patches K is 8, and the loss balance factor β is 0.2. In addition, we use GPT-5.4 to generate class-wise attribute semantics, and randomly sample $N = 5$ attribute descriptions for each class. Results with other LLMs are provided in the appendix.

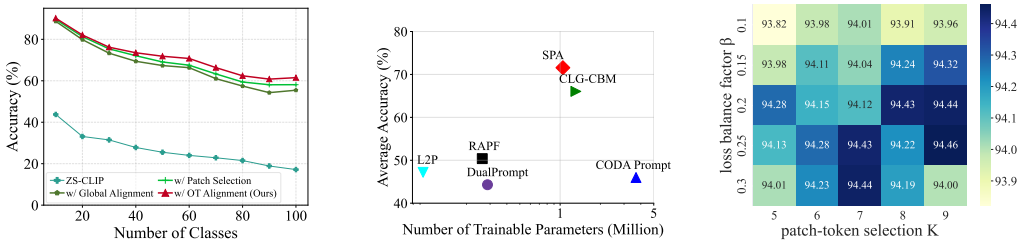
Evaluation metric. Following the common CIL protocol (Rebuffi et al., 2017), we evaluate all methods using the last accuracy $\mathcal{A}_B$ and the average accuracy $\bar{\mathcal{A}} = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}_b$, where $\mathcal{A}_b$ denotes the test accuracy after the b-th incremental session. Specifically, $\mathcal{A}_B$ is the accuracy over all learned classes after the final session. $\bar{\mathcal{A}}$ denotes the average accuracy across all sessions, measuring the overall performance and stability during incremental learning.

5.2 Benchmark Comparison

We first compare SPA with SOTA CIL methods on nine benchmark datasets, and report the results in Table 1 and Fig. 3. Overall, SPA consistently achieves the best performance, demonstrating the effectiveness of patch-level alignment for CLIP-based CIL. Visual prompt-based methods such as L2P and CODA-Prompt perform less competitively, as they cannot exploit rich textual semantics. CLIP-based methods such as RAPF and CLG-CBM achieve better performance by leveraging CLIP's cross-modal representation capability. However, these methods still mainly rely on global alignment. In contrast, SPA introduces semantic-guided patch selection and structured patch-level alignment, thereby learning more discriminative representations and mitigating forgetting.

5.3 Further Analysis

Ablation study. We conduct ablation studies to analyze the contribution of each component in SPA on Aircraft B0 Inc10, and report the results in Fig. 4a. Specifically, we take **"ZS-CLIP"** as the baseline, whose performance is relatively limited. Based on this, we first introduce global alignment (Eq. 13), denoted as **"w/ Global Alignment"**, which brings clear performance gains and demon-



(a) Ablation study (b) Trainable parameters (c) Parameter sensitivity

Figure 4: Ablation study, trainable parameters, and parameter sensitivity.



Figure 5: **Column 1:** Input. **Column 2:** Response heatmap of discriminative patches. **Column 3:** Visualization of the top-8 discriminative patches selected. **Column 4:** OT-based correspondence graph, where numbered nodes denote the selected patches and attribute nodes denote class-wise attributes. For clarity, only the top-3 correspondences with the highest transport weights are shown.

strates the effectiveness of adapting CLIP to downstream incremental learning. We further introduce discriminative patch selection, denoted as **"w/ Patch Selection"**, which selects discriminative patches under the guidance of class-wise semantics (Eq. 7), and aligns them with these semantics via a simple matching strategy. The additional gain shows that filtering irrelevant local regions improves the effectiveness of local representations. Finally, we introduce optimal transport-based structured alignment (Eq. 12), denoted as **"w/ OT Alignment"**, which gives to the full SPA and achieves the best performance. This further demonstrates the effectiveness of balanced matching.

Trainable parameters. We compare the trainable parameters and average accuracy on Aircraft B0 Inc10 in Fig. 4b. The trainable parameters of SPA mainly come from the lightweight task-specific projectors. Compared with prompt-based methods, *e.g.*, L2P, SPA introduces more parameters but achieves higher accuracy. Compared with CLIP-based methods, SPA achieves the best performance with a moderate parameter count, showing a favorable accuracy-parameter trade-off.

Parameter robustness. We analyze the sensitivity of our method on Cars B0 Inc10 by varying two key hyperparameters: the number of selected patches K and the loss balance factor β. Specifically, we set $K \in \{5, 6, 7, 8, 9\}$ and $\beta \in \{0.1, 0.15, 0.2, 0.25, 0.3\}$. We report the average performance in Fig. 4c. As shown in the figure, the performance remains relatively stable across different choices of K and β. A proper patch number can provide a stable trade-off between preserving informative local information and suppressing redundant noise.

Visualizations. Fig. 5 visualizes the proposed OT-based patch-level alignment. The selection heatmaps (**Column 2**) show that class-wise attributes guide the model to focus on discriminative local regions. The OT-based correspondence graphs (**Column 4**) further reveal structured associations between selected patches and class-wise semantic attributes. For example, the attribute *"very narrow, elongated head"* describes the bird head and is assigned a strong correspondence to patch *4* on the head region. These results show that SPA can build meaningful cross-modal associations between local visual regions and semantic attributes. More results are provided in the Appendix.

6 Conclusion

We propose SPA, a semantic-guided patch-level alignment framework for CLIP-based CIL. Different from previous methods that mainly rely on global alignment, SPA explicitly exploits the rich local representations in CLIP. SPA first constructs diverse visual samples for each class, and then leverages GPT to generate class-wise semantics. These semantics are used to select discriminative patches and align them with textual attributes through optimal transport, while task-specific projectors are introduced to mitigate catastrophic forgetting. Extensive experiments demonstrate that SPA achieves strong performance across multiple benchmarks.

Limitations and Future Work. SPA relies on externally generated attribute semantics, whose quality may affect patch selection and local alignment. In addition, optimal transport introduces extra computation. Future work will explore more efficient and robust patch-level alignment strategies.

AI USE STATEMENT

In this work, we used generative AI tools for research execution and manuscript preparation. Specifically, GPT-5.4 was used to generate class-wise visual attribute semantics from selected representative and diverse samples, which are incorporated into SPA for semantic-guided patch selection and local cross-modal alignment. Additionally, we used generative AI tools to check language, notation consistency, and presentation clarity throughout the manuscript. We reviewed and verified all AI-assisted work and take responsibility for the final content of this work, including the experimental results, claims, and conclusions.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 139–154, 2018.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2(1):1, 2023.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

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

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Takuma Fukuda, Hiroshi Kera, and Kazuhiko Kawamoto. Adapter merging with centroid prototype mapping for scalable class-incremental learning. In *Proceedings of the computer vision and pattern recognition conference*, pp. 4884–4893, 2025.

Wilfrid Gangbo and Robert J McCann. The geometry of optimal transportation. 1996.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International journal of computer vision*, 132(2):581–595, 2024.

Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

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

Tao Hu, Lan Li, Zhen-Hao Xie, and Da-Wei Zhou. Hierarchical semantic tree anchoring for clip-based class-incremental learning. *arXiv preprint arXiv:2511.15633*, 2025.

Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *European Conference on Computer Vision*, pp. 214–231. Springer, 2024.

Linlan Huang, Xusheng Cao, Haori Lu, Yifan Meng, Fei Yang, and Xialei Liu. Mind the gap: Preserving and compensating for the modality gap in clip-based continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3777–3786, 2025.

Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, et al. Openclip. *Zenodo*, 2021.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

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

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Marc Lafon, Elias Ramzi, Clément Rambour, Nicolas Audebert, and Nicolas Thome. Gallop: Learning global and local prompts for vision-language models. In *European Conference on Computer Vision*, pp. 264–282. Springer, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Lan Li, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Addressing imbalanced domain-incremental learning through dual-balance collaborative experts. *arXiv preprint arXiv:2507.07100*, 2025.

Lan Li, Tao Hu, Da-Wei Zhou, Jia-Qi Yang, Han-Jia Ye, and De-Chuan Zhan. Bofa: Bridge-layer orthogonal low-rank fusion for clip-based class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22967–22975, 2026.

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

Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 2544–2553, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Zilin Luo, Yaoyao Liu, Bernt Schiele, and Qianru Sun. Class-incremental exemplar compression for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11371–11380, 2023.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11321–11329, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Zhi-Hong Qi, Da-Wei Zhou, Yiran Yao, Han-Jia Ye, and De-Chuan Zhan. Adaptive adapter routing for long-tailed class-incremental learning. *Machine Learning*, 114(3):68, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. *Advances in neural information processing systems*, 34:6747–6761, 2021.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11909–11919, 2023.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Hao Sun and Da-Wei Zhou. C3box: A clip-based class-incremental learning toolbox. *arXiv preprint arXiv:2601.20852*, 2026.

Yuwen Tan, Qinhao Zhou, Xiang Xiang, Ke Wang, Yuchuan Wu, and Yongbin Li. Semantically-shifted incremental adapter-tuning is a continual vitransformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23252–23262, 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Fu-Yun Wang, Da-Wei Zhou, Liu Liu, Han-Jia Ye, Yatao Bian, De-Chuan Zhan, and Peilin Zhao. Beef: Bi-compatible class-incremental learning via energy-based expansion and fusion. In *The eleventh international conference on learning representations*, 2022a.

Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022b.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pp. 631–648. Springer, 2022c.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022d.

Aoqi Wu, Yifan Yang, Xufang Luo, Yuqing Yang, Chunyu Wang, Liang Hu, Xiyang Dai, Dongdong Chen, Chong Luo, Lili Qiu, et al. Llm2clip: Powerful language model unlock richer visual representation. In *NeurIPS 2024 Workshop: Self-Supervised Learning-Theory and Practice*, 2024a.

Mingrui Wu, Xinyue Cai, Jiayi Ji, Jiale Li, Oucheng Huang, Hao Fei, Guannan Jiang, Xiaoshuai Sun, and Rongrong Ji. Controlmllm: Training-free visual prompt learning for multimodal large language models. *Advances in Neural Information Processing Systems*, 37:45206–45234, 2024b.

Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. Incremental learning using conditional adversarial networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6619–6628, 2019.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Ju Xu and Zhanxing Zhu. Reinforced continual learning. *Advances in neural information processing systems*, 31, 2018.

Yuan Yao, Qianyu Chen, Ao Zhang, Wei Ji, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Pevl: Position-enhanced pre-training and prompt tuning for vision-language models. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 11104–11117, 2022.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23219–23230, 2024.

Lu Yu, Haoyu Han, Zhe Tao, Hantao Yao, and Changsheng Xu. Language guided concept bottleneck models for interpretable continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14976–14986, 2025.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13208–13217, 2020.

Bowen Zheng, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Task-agnostic guided feature expansion for class-incremental learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10099–10109, 2025.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: a survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 8363–8371, 2024a.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024b.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3):1012–1032, 2025a.

Da-Wei Zhou, Kai-Wen Li, Jingyi Ning, Han-Jia Ye, Lijun Zhang, and De-Chuan Zhan. External knowledge injection for clip-based class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3314–3325, 2025b.

Da-Wei Zhou, Yuanhan Zhang, Yan Wang, Jingyi Ning, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Learning without forgetting for vision-language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025c.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5871–5880, 2021.

APPENDIX

In this supplementary material, we provide more details about SPA, including more implementation details and experimental results. The appendix is organized as follows:

Appendix. A provides additional experimental analyses, including computational efficiency, backbone robustness, multiple random class orders, LLM variants, shared GPT-generated semantics, top-K patch selection strategies, parameter robustness, and forgetting analysis.

Appendix. B provides visualizations of our method, including discriminative patch selection, OT-based patch-level alignment, and examples of generated class-wise attribute semantics.

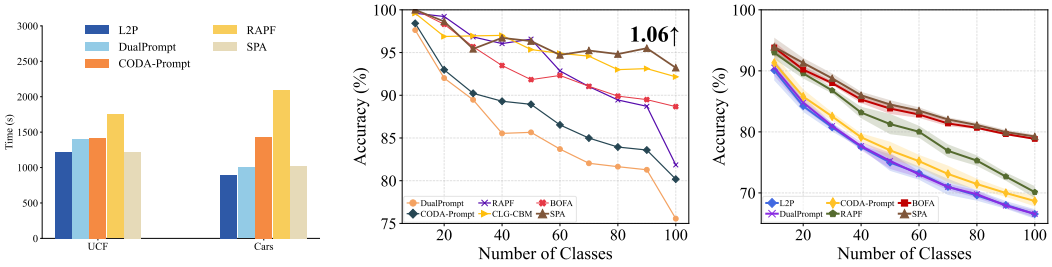
Appendix. C describes the training algorithm of our method.

Appendix. D introduces the compared methods used in experiments.

Appendix. E reports full results across datasets and incremental settings.

A MORE RESULTS

In this section, we provide additional experimental analyses to complement the main paper. We first evaluate the computational efficiency of different methods, examine SPA under different CLIP pre-trained weights, and report results with multiple random class orders. We then analyze the influence of different LLMs, shared GPT-generated semantics, trainable parameters, and top-K patch selection strategies.



(a) Running time comparison (b) OpenAI pre-trained weight (c) Multiple runs

Figure 6: Running time comparison, OpenAI pre-trained weight results, and multiple runs.

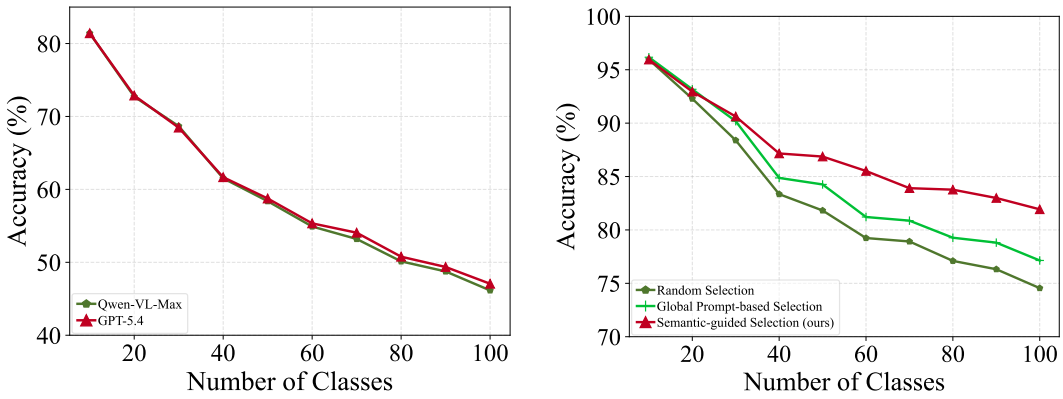
A.1 COMPUTATIONAL EFFICIENCY

Overall Running Time. We further compare the running time of different methods under the same experimental setting. All experiments are conducted on a single NVIDIA 4090 GPU. As shown in Fig. 6a, SPA requires less running time than RAPF and CODA-Prompt, and is comparable to prompt-based methods such as L2P and DualPrompt. These results show that SPA achieves strong performance without introducing excessive computational overhead.

Inference Cost. The running time reported in Fig. 6a includes the complete training and inference cost of SPA. Note that GPT-5.4 is used only for offline generation of class-wise attribute semantics. Once generated, these semantics are stored and directly reused during inference, without any further GPT-5.4 calls. However, since SPA additionally performs semantic-guided patch selection and OT-based local alignment during inference, we further analyze its inference overhead under the same experimental setting. As shown in Table 2, SPA achieves comparable inference latency and moderate GPU memory consumption compared with representative CLIP-based CIL methods. We further analyze the scalability of SPA as the number of seen classes increases on SUN. As shown in Table 3, the overall inference latency grows moderately, mainly due to the local branch, while the Sinkhorn

Table 2: Comparison of inference efficiency on ImageNet-R B0 Inc20.

Method	$\bar{A}$	Latency (ms/img)	Memory (MB)
CLG-CBM (Yu et al., 2025)	84.64	**4.73**	**2342**
BOFA (Li et al., 2026)	84.53	5.34	2900
SPA (Ours)	**85.63**	5.16	2583



(a) Different LLMs

(b) Top-K patch selection strategies

Figure 7: Different LLMs, and Top-K patch selection strategies.

solver contributes only a small fraction of the total cost. These results indicate that SPA introduces limited practical overhead and scales reasonably well to larger class-incremental settings.

A.2 Different Backbones

In the main paper, we report results using LAION-400M pre-trained CLIP (Ilharco et al., 2021). To further evaluate the generality of SPA, we also conduct experiments with OpenAI pre-trained CLIP (Radford et al., 2021) on UCF B0 Inc10. As shown in Fig. 6b, SPA consistently outperforms compared methods under different pre-trained weights. This indicates that our method is not tied to a specific CLIP initialization and can bring consistent performance gains under different CLIP pre-training settings.

Table 3: Scalability analysis of SPA with increasing seen classes of on SUN B0 Inc30.

Seen Classes	Inference Latency (ms/img)	Local Branch (ms/img)	Sinkhorn (ms/img)
120	4.64	1.74	0.09
210	5.24	2.38	0.07
300	6.64	3.63	0.07

A.3 Multiple Runs

To evaluate the robustness of SPA, we conduct experiments with multiple random class orders on ImageNet-R B0 Inc20. Specifically, we use five random seeds, *i.e.*, 1993, 1994, 1995, 1996, and 1997, to generate different class orders and report the average performance with standard deviation. As shown in Fig. 6c, SPA consistently achieves strong performance across different runs, showing that it is robust to variations in class order.

A.4 Different LLMs

We further investigate the influence of different LLMs for generating class-wise attribute semantics. Specifically, for fair comparison, we compare GPT-5.4 (Singh et al., 2025) with Qwen-VL-Max (Bai et al., 2023) under the same visual samples and prompt template on ObjectNet B0 Inc20. As shown in Fig. 7a, SPA achieves stable performance with both LLMs, indicating that the proposed semantic-guided patch selection is not limited to a specific LLM.

A.5 Top-K Patch Selection Strategies

We further compare different top-K patch selection strategies on CUB B0 Inc20 to verify the effectiveness of our semantic-guided patch selection. **Random Selection** randomly selects K patches from all image patches without using any semantic guidance, while **Global Prompt-based Selection** ranks all patches according to their similarity to the class-name prompt (*e.g.*, "a photo of a [CLASS]") and selects the top-K patches. In contrast, our method (*i.e.*, **Semantic-guided Selection**) selects patches based on their relevance to class-wise attribute semantics. As shown in Fig. 7b,

Table 4: Fair comparison when all methods use the same GPT-generated attribute semantics. Results are reported as *mean* ± *standard* deviation over five random seeds. The best results are highlighted in bold.

Method	Food B0 Inc10		UCF B0 Inc10		ObjectNet B0 Inc20	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
CLG-CBM (Yu et al., 2025)	88.71±0.52	83.49±0.26	94.67±0.62	91.45±0.34	55.29±1.98	44.72±0.28
PROOF (Zhou et al., 2025c)	88.80±0.76	83.40±0.82	93.82±0.76	90.83±0.25	53.42±1.62	44.09±0.37
BOFA (Li et al., 2026)	87.88±0.51	82.21±0.12	92.96±0.64	88.78±0.10	56.87±1.63	47.02±0.08
SPA (Ours)	**89.70±0.70**	**84.31±0.31**	**95.81±0.37**	**93.55±0.41**	**58.25±1.92**	**47.63±0.43**

our strategy achieves the best performance, showing that attribute semantics can better identify discriminative local regions and reduce irrelevant background noise.

A.6 Effect of GPT-generated Semantics

To examine whether the gains of SPA mainly come from stronger GPT-generated semantic priors, we provide CLG-CBM (Yu et al., 2025), PROOF (Zhou et al., 2025c), and BOFA (Li et al., 2026) with the same GPT-5.4 attribute semantics used by SPA. For a fair comparison, we repeat the experiments over five random seeds *i.e.*, 1993, 1994, 1995, 1996, and 1997, on Food101 B0 Inc10, UCF B0 Inc10, and ObjectNet B0 Inc20, and report the mean and standard deviation across the five runs. As shown in Table 4, SPA consistently outperforms the compared methods across all three datasets when equipped with identical GPT-5.4 attribute semantics. This suggests that the performance gains of SPA cannot be attributed solely to stronger semantic priors, and that the proposed discriminative patch selection and OT-based local alignment contribute additional improvements under the same semantic guidance.

A.7 Parameter Robustness

For the parameter analysis reported in the main paper, we further verify the trends on a validation split. Specifically, we split the original training data into a training subset and a validation subset with a ratio of $4:1$. We conduct hyperparameter sensitivity analysis on the validation subset, and observe trends consistent with those reported in the main paper. These results further support the robustness of SPA to different hyperparameter settings.

A.8 Forgetting Measure

We evaluate the forgetting degree of different methods using F_B, which measures how much the performance on previous tasks decreases after learning subsequent tasks. Formally, it is defined as:

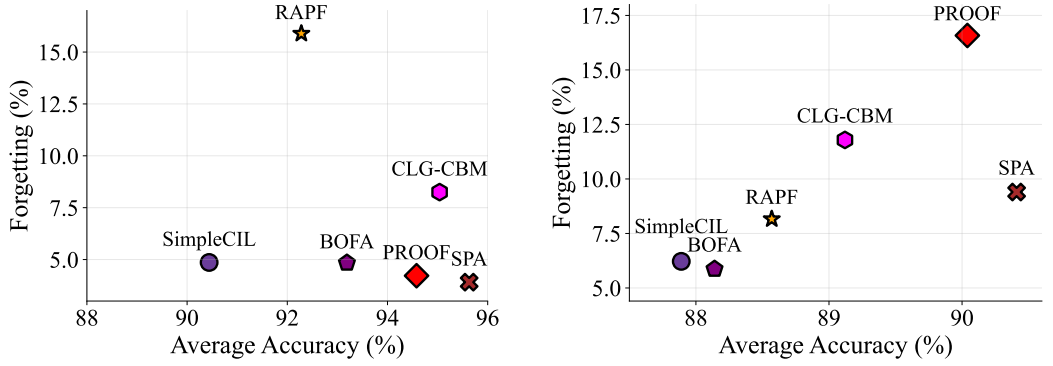
$$F_B = \frac{1}{B-1} \sum_{b=1}^{B-1} \max_{l \in \{b,\dots,B-1\}} (\mathcal{A}_{l,b} - \mathcal{A}_{B,b}), \tag{16}$$

where $\mathcal{A}_{l,b}$ denotes the accuracy on task b after learning stage l. A smaller F_B indicates better preservation of previously learned knowledge.

As shown in Table 5, SPA maintains relatively low forgetting across different incremental settings, achieving the lowest F_B on UCF B0 Inc10 and both SUN settings, while remaining competitive on Cars. Fig. 8 further illustrates the trade-off between average accuracy and forgetting, where the lower-right region indicates higher accuracy with less forgetting. On UCF B0 Inc10, SPA is located near this desirable region, achieving the highest average accuracy with the lowest forgetting among compared methods. On Food B0 Inc10, methods such as SimpleCIL and BOFA exhibit relatively small forgetting, but they achieve lower average accuracy than SPA, suggesting a stronger emphasis on stability than plasticity. In contrast, SPA achieves the highest average accuracy while maintaining a competitive forgetting measure, demonstrating a favorable stability-plasticity trade-off.

Table 5: Forgetting measure F_B of different methods under different incremental settings. The best results are highlighted in bold. Lower F_B indicates less forgetting.

Method	Food		Cars		UCF		SUN	
	B0 Inc10	B50 Inc10	B0 Inc10	B50 Inc10	B0 Inc10	B50 Inc10	B0 Inc30	B150 Inc30
ZS-CLIP (Radford et al., 2021)	**5.22**	**2.84**	6.41	3.14	10.77	4.74	10.16	4.97
SimpleCIL (Zhou et al., 2025a)	6.22	3.76	5.11	2.51	4.86	**2.31**	8.12	3.51
L2P (Wang et al., 2022d)	24.69	20.26	7.45	4.22	8.15	6.22	22.87	17.57
DualPrompt (Wang et al., 2022c)	24.28	20.77	7.35	4.71	9.26	7.95	23.54	17.63
CODA-Prompt (Smith et al., 2023)	25.82	22.41	6.14	4.59	9.80	9.21	22.71	17.69
RAPF (Huang et al., 2024)	8.16	7.83	28.27	29.30	15.89	20.50	14.58	9.36
CLG-CBM (Yu et al., 2025)	11.79	9.38	6.86	4.86	8.25	6.87	14.62	12.26
PROOF (Zhou et al., 2025c)	16.58	16.94	4.48	2.30	4.22	4.85	18.33	15.71
BOFA (Li et al., 2026)	5.87	4.08	**3.94**	**2.17**	4.84	4.13	8.47	4.16
SPA (Ours)	9.40	7.93	4.66	2.22	**3.91**	4.33	**6.71**	**3.31**



(a) UCF B0 Inc10

(b) Food B0 Inc10

Figure 8: Accuracy-forgetting trade-off on UCF B0 Inc10 and Food B0 Inc10. The lower-right region indicates higher average accuracy and lower forgetting.

B Visualizations

B.1 More Visualizations

Fig. 9 provides additional visualizations of the proposed semantic-guided patch selection and OT-based patch-attribute alignment. Across different categories, the response heatmaps show that SPA consistently focuses on category-relevant local regions, such as the ears and striped coat of the *cat*, the head and muzzle of the *whippet*, and the clustered arils of the *pomegranate*. In the OT-based correspondence graphs, semantic attributes are structurally associated with their corresponding visual regions. For example, the attribute "upright, triangular ears" is linked to patches on the *cat ears*, while "numerous small, round red arils" is associated with patches covering the *pomegranate seeds*. These results further demonstrate that SPA can establish meaningful token-level relations between local visual patches and textual attributes.

B.2 Generated class-wise attribute semantics

In this section, we present examples of class-wise attribute semantics generated the by GPT-5.4 (Singh et al., 2025) on ImageNet-R:

duck:

> **Q**: *What are the key visual features for identifying a duck in these images? Focus on the most discriminative attributes.*
> **A**: **1.** *Relatively short neck connecting the head and body.* **2.** *Short, upturned tail at the rear of the body.* **3.** *Paired wings along the sides.* **4.** *Bills are often colored orange or red.* **5.** *Webbed feet are visible beneath the body.* **6.**$\cdots$

snail:

> **Q**: *What are the key visual features for identifying a snail in these images? Focus on the most discriminative attributes.*

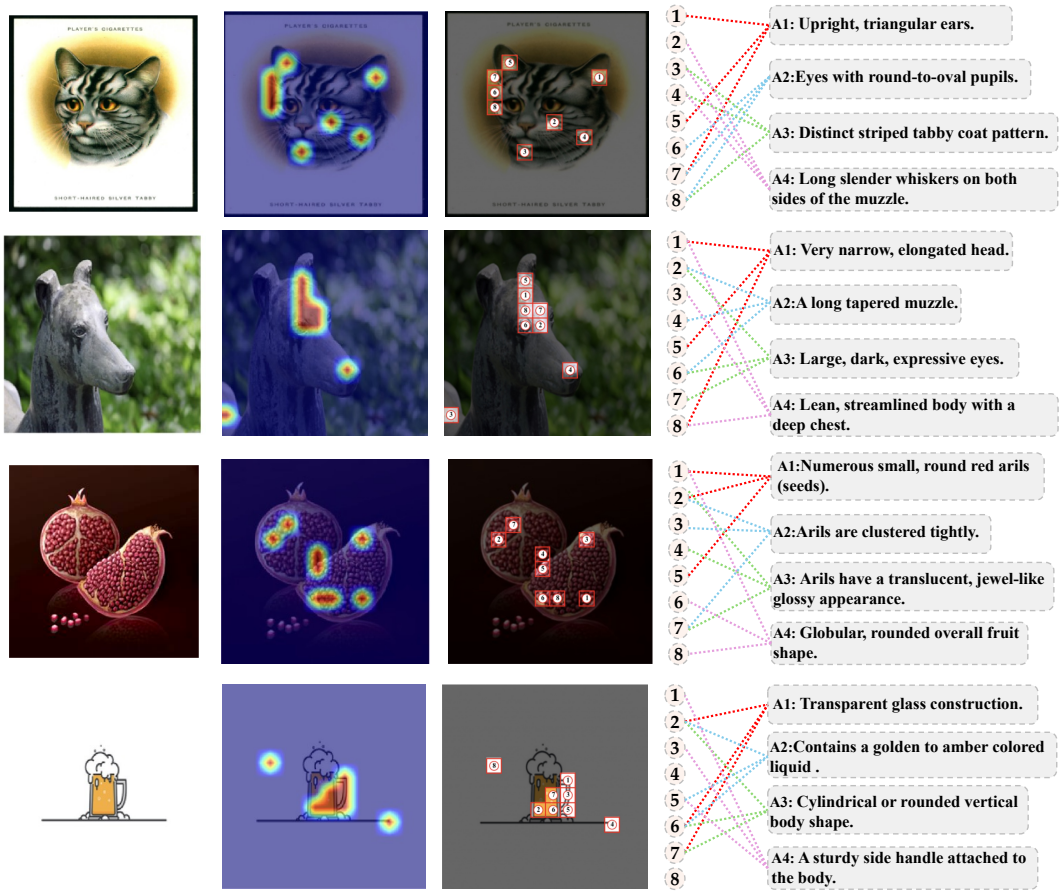


Figure 9: **Column 1:** Input. **Column 2:** Response heatmap of discriminative patches. **Column 3:** Visualization of the top-8 discriminative patches selected. **Column 4:** OT-based correspondence graph, where numbered nodes denote the selected patches and attribute nodes denote class-wise attributes. For clarity, only the top-3 correspondences with the highest transport weights are shown.

Algorithm 1 Training SPA for CIL

Input: Current incremental task dataset $\mathcal{D}^b$; Current model: $f(\cdot)$;
Output: Updated model;

Freeze all previous global and local projectors;
Initialize current global and local projectors;
for each new class c in $\mathcal{D}^b$ **do**
  Extract global visual features via $\bar{g}_i(\cdot)$;
  Compute the class prototype $\mathbf{p}_c$ via Eq. 3;
  Construct representative and diverse visual samples $\mathcal{S}_c$ via Eqs. 4 and 5;
  Generate class-wise attribute semantics $\mathcal{A}_c$ with GPT-5.4 based on $\mathcal{S}_c$;
  Encode A_c into textual embeddings $\mathcal{T}_c$ via $\bar{g}_t(\cdot)$;
end for
for $(\mathbf{x}, y) \in \mathcal{D}^b$ **do**
  Extract $\mathbf{v}_{\mathrm{cls}}$ and patch-level features $\mathcal{V}$ via $\bar{g}_i(\mathbf{x})$;
  Extract global text embeddings $\mathbf{t}^y_{\mathrm{eos}}$ via $\bar{g}_t(\cdot)$;
  Compute adapted global and local features with task-specific projectors via Eq. 8;
  Select top-K discriminative patches guided by $\mathcal{T}_y$ via Eq. 7;
  Perform patch-level alignment via optimal transport in Eq. 10;
  Compute the global loss $\mathcal{L}_g$ via Eq. 13;
  Compute the local loss $\mathcal{L}_l$ via Eq. 12;
  Compute the total loss $\mathcal{L}$ via Eq. 14; update the model;
end for
return the updated model.

> **A***: **1.** Distinct rounded coiled shell with a clear spiral pattern.* **2.** *Soft, elongated flattened foot extends beneath the shell.* **3.** *The shell is proportionally large.* **4.** *A head bearing two long upper tentacles.* **5.** *Shells show earthy or golden hues, while the soft body is darker.* **6.**$\cdots$

shih_tzu:

> **Q***: What are the key visual features for identifying a shih_tzu in these images? Focus on the most discriminative attributes.*
> **A***: **1.** Dense, long, silky coat that covers the body and head.* **2.** *Short, pushed-in muzzle with a broad black nose.* **3.** *Broad, rounded skull producing a domed forehead.* **4.** *Long, heavily coated ears that hang down alongside the face.* **5.** *Small, compact body with short legs.* **6.**$\cdots$

pig:

> **Q***: What are the key visual features for identifying a pig in these images? Focus on the most discriminative attributes.*
> **A***: **1.** A short, curly or corkscrew-shaped tail.* **2.** *Short, rounded head with a broad snout area.* **3.** *Distinct round, flattened snout with two visible nostrils.* **4.** *Small eyes set on the sides of the head.* **5.** *Stout, barrel-shaped body that is broad and compact.* **6.**$\cdots$

mobile_phone:

> **Q***: What are the key visual features for identifying a mobile_phone in these images? Focus on the most discriminative attributes.*
> **A***: **1.** Thin, flat rectangular handheld shape.* **2.** *Prominent rectangular display that occupies much of the front face.* **3.** *Corners and edges are generally rounded.* **4.** *Small enough to be held and operated with a single hand.* **5.** *Largely featureless aside from the screen and buttons.* **6.**$\cdots$

C Overview of Training Algorithm

Table 6: Comparison of the average and last performance of different methods. The best results are highlighted in bold. **All methods are initialized from the same pre-trained CLIP backbone.**

Method	Aircraft				CIFAR100				Cars			
	B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
ZS-CLIP (Radford et al., 2021)	26.66	17.22	21.70	17.22	81.81	71.38	76.49	71.38	82.60	76.37	78.32	76.37
SimpleCIL (Zhou et al., 2025a)	59.24	48.09	53.05	48.09	84.15	76.63	80.20	76.63	92.04	86.85	88.96	86.85
L2P (Wang et al., 2022d)	47.19	28.29	44.07	32.13	82.74	73.03	81.14	73.61	76.63	61.82	76.37	65.64
DualPrompt (Wang et al., 2022c)	44.30	25.83	46.07	33.57	81.63	72.44	80.12	72.57	76.26	62.94	76.88	67.55
CODA-Prompt (Smith et al., 2023)	45.98	27.69	45.14	32.28	82.43	73.43	78.69	71.58	80.21	66.47	75.06	64.19
RAPF (Huang et al., 2024)	50.38	23.61	40.47	25.44	86.14	78.04	82.17	77.93	82.89	62.85	75.87	63.19
CLG-CBM (Yu et al., 2025)	66.05	55.93	59.25	55.39	86.58	80.15	83.59	79.28	93.25	88.76	90.11	88.19
PROOF (Zhou et al., 2025c)	63.81	56.14	59.47	57.10	86.77	79.11	83.32	79.73	90.74	86.51	88.00	85.58
BOFA (Li et al., 2026)	70.96	60.43	66.09	61.36	86.07	79.19	83.02	79.44	94.21	90.20	92.13	90.50
SPA (Ours)	**71.57**	**61.51**	**66.82**	**63.01**	**88.53**	**81.81**	**85.01**	**81.60**	**94.43**	**90.91**	**92.33**	**91.43**

Method	ImageNet-R				CUB				UCF			
	B0 Inc20		B100 Inc20		B0 Inc20		B100 Inc20		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
ZS-CLIP (Radford et al., 2021)	83.37	77.17	79.57	77.17	74.38	63.06	67.96	63.06	75.50	67.64	71.44	67.64
SimpleCIL (Zhou et al., 2025a)	81.06	74.48	76.84	74.48	83.81	77.52	79.75	77.52	90.44	85.68	88.12	85.68
L2P (Wang et al., 2022d)	75.97	66.52	72.82	66.77	70.87	57.93	75.64	66.12	86.34	76.43	83.95	76.62
DualPrompt (Wang et al., 2022c)	76.21	66.65	73.22	67.58	69.89	57.46	74.40	64.84	85.21	75.82	84.31	76.35
CODA-Prompt (Smith et al., 2023)	77.69	68.95	73.71	68.05	73.12	62.98	73.95	62.21	87.76	80.14	83.04	75.03
RAPF (Huang et al., 2024)	81.26	70.48	76.10	70.23	79.09	62.77	72.82	62.93	92.28	80.33	90.31	81.55
CLG-CBM (Yu et al., 2025)	84.64	78.50	81.46	77.88	85.37	78.24	77.74	76.97	95.04	91.36	94.17	91.85
PROOF (Zhou et al., 2025c)	83.84	78.40	81.20	78.92	82.31	76.64	79.20	76.37	94.58	91.10	93.58	90.91
BOFA (Li et al., 2026)	84.53	78.77	81.60	79.12	86.66	80.58	83.18	80.79	93.19	88.71	92.60	89.43
SPA (Ours)	**85.63**	**79.08**	**82.50**	**79.50**	**87.17**	**81.93**	**84.43**	**82.23**	**95.63**	**92.38**	**95.43**	**93.48**

Method	SUN				Food				ObjectNet			
	B0 Inc30		B150 Inc30		B0 Inc10		B50 Inc10		B0 Inc20		B100 Inc20	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
ZS-CLIP (Radford et al., 2021)	79.42	72.11	74.95	72.11	87.86	81.92	84.75	81.92	38.43	26.43	31.12	26.43
SimpleCIL (Zhou et al., 2025a)	82.13	75.58	78.62	75.58	87.89	81.65	84.73	81.65	52.06	40.13	45.11	40.13
L2P (Wang et al., 2022d)	82.82	74.54	79.57	73.10	85.66	77.33	80.42	73.13	51.40	39.39	48.91	42.83
DualPrompt (Wang et al., 2022c)	82.46	74.40	79.37	73.02	84.92	77.29	80.00	72.75	52.62	40.72	49.08	42.92
CODA-Prompt (Smith et al., 2023)	83.34	75.71	80.38	74.17	86.18	78.78	80.98	74.13	46.49	34.13	40.57	34.13
RAPF (Huang et al., 2024)	82.13	72.47	78.04	73.10	88.57	81.15	85.53	81.17	48.67	27.43	39.28	28.73
CLG-CBM (Yu et al., 2025)	84.85	78.09	81.58	77.79	89.12	83.05	86.76	83.85	58.53	45.11	49.80	43.56
PROOF (Zhou et al., 2025c)	83.89	77.25	80.15	76.54	90.04	84.73	87.52	84.74	56.07	43.69	48.90	43.62
BOFA (Li et al., 2026)	84.38	77.60	81.34	77.93	88.14	82.08	85.97	82.84	59.21	46.95	51.89	46.76
SPA (Ours)	**86.91**	**80.77**	**83.57**	**80.74**	**90.41**	**84.81**	**87.80**	**84.95**	**59.98**	**47.06**	**53.28**	**47.59**

The training process of SPA is detailed in Algorithm 1. At the beginning of the b-th incremental task, we freeze the previously learned visual and textual projectors for both global and local branches, and initialize the current ones (Line 1 to Line 2). For each new class $c \in Y_b$, SPA first computes the prototype $\mathbf{p}_c$ (Line 5), constructs the representative and diverse visual sample set $\mathcal{S}_c$ (Line 6), and generates class-wise attribute semantics A_c with GPT-5.4 based on $\mathcal{S}_c$ (Line 7). During training, we extract the global visual feature $\mathbf{v}_{\mathrm{cls}}$, patch-level feature set $\mathcal{V}$, and global text embeddings $\mathbf{t}^y_{\mathrm{eos}}$ using frozen CLIP encoders (Line 11 to Line 12). Then, the current projectors and frozen historical projectors are cumulatively applied to obtain adapted global and local features. Based on the adapted features (Line 13), SPA selects top-K discriminative patches guided by $\mathcal{T}_y$ (Line 14), which are further aligned with textual attributes via optimal transport (Line 15). Finally, SPA jointly optimizes the global loss $\mathcal{L}_g$ and local loss $\mathcal{L}_l$ to update only the current task projectors (Line 16 to Line 18). The final updated model is then returned as the output of the training process.

D Introduction About Compared Methods

In this section, we briefly introduce the compared methods used in our experiments. All methods are initialized from the same pre-trained CLIP backbone.

ZS-CLIP (Radford et al., 2021): This baseline directly performs prediction with the frozen CLIP without any training or parameter updates. It serves as a zero-shot performance reference for pre-trained CLIP in the incremental setting.

SimpleCIL (Zhou et al., 2025a): This baseline freezes the pre-trained visual encoder and performs classification with the extracted visual representations, without using the text encoder. It provides a simple yet strong reference for CIL.

L2P (Wang et al., 2022d): This method learns the prompt pool and retrieves suitable prompts according to the input image to adapt a frozen visual backbone. Since it is designed for visual prompt

learning, it does not exploit the CLIP text encoder or textual semantics.

DualPrompt (Wang et al., 2022c): Built upon L2P, this method introduces two types of prompts, *i.e.*, general prompts and expert prompts, to capture task-shared and task-specific knowledge, respectively. Similar to L2P, it only adapts the visual branch and does not use the text encoder.

CODA-Prompt (Smith et al., 2023): This method learns decomposed attention-based prompts and dynamically composes prompts for each instance. It only uses the visual branch of CLIP.

RAPF (Huang et al., 2024): This method is a CLIP-based CIL approach that leverages textual knowledge to adaptively adjust class prototypes and further applies parameter fusion to reduce interference between incremental tasks.

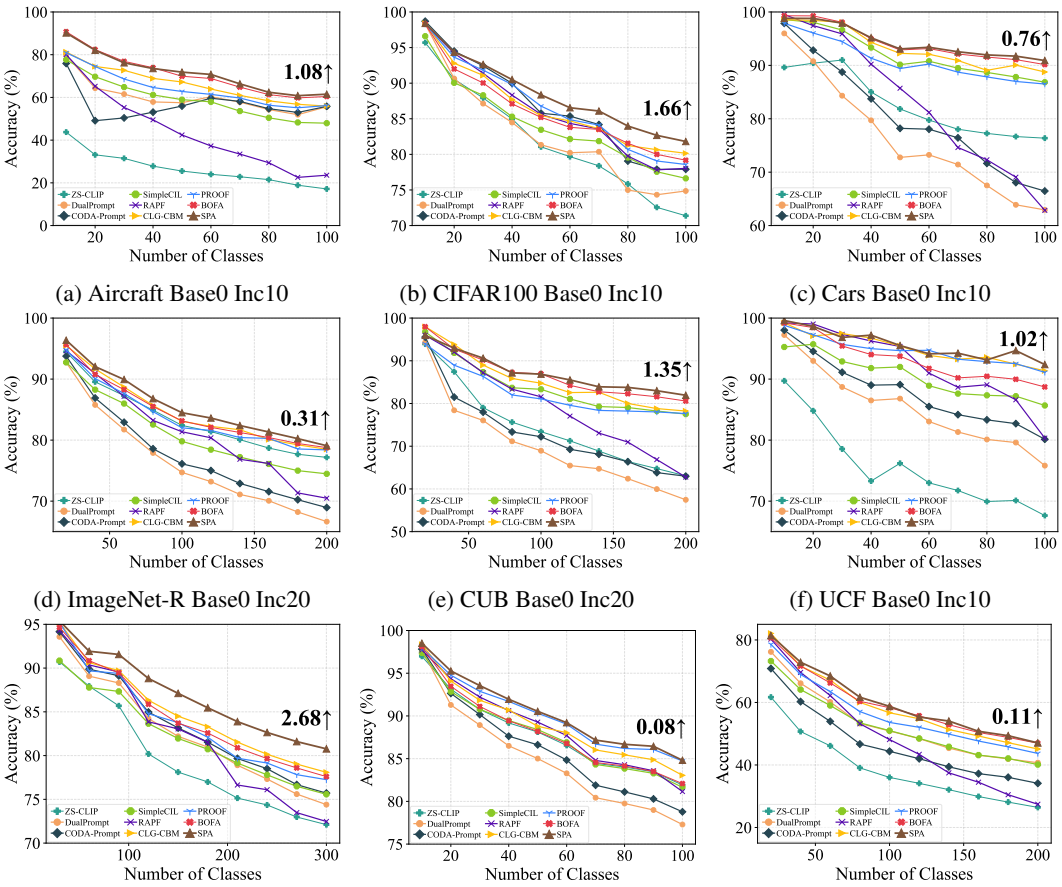
CLG-CBM (Yu et al., 2025): This method builds a language-guided concept bottleneck model for continual learning. It maps visual representations into interpretable concept spaces and leverages textual concepts to improve both recognition performance and prediction interpretability.

PROOF (Zhou et al., 2025c): This method introduces task-specific projection layers on frozen CLIP encoders and a cross-modal fusion layer to integrate visual and textual knowledge. By aggregating the learned projections across incremental tasks, it adapts CLIP to new classes while preserving previously acquired knowledge.

BOFA (Li et al., 2026): This method proposes bridge-layer orthogonal low-rank fusion for CLIP-based CIL. It introduces lightweight low-rank updates and constrains them to reduce interference between old and new tasks, improving the stability of incremental adaptation.

E FULL RESULTS

In this section, we provide the complete incremental performance results of different methods in Table 6. We report the incremental performance curves under the B0 setting in Fig. 10 and the half-base setting in Fig. 11. As shown in these results, SPA consistently maintains strong performance across different datasets and data splits, demonstrating its effectiveness in CLIP-based CIL.

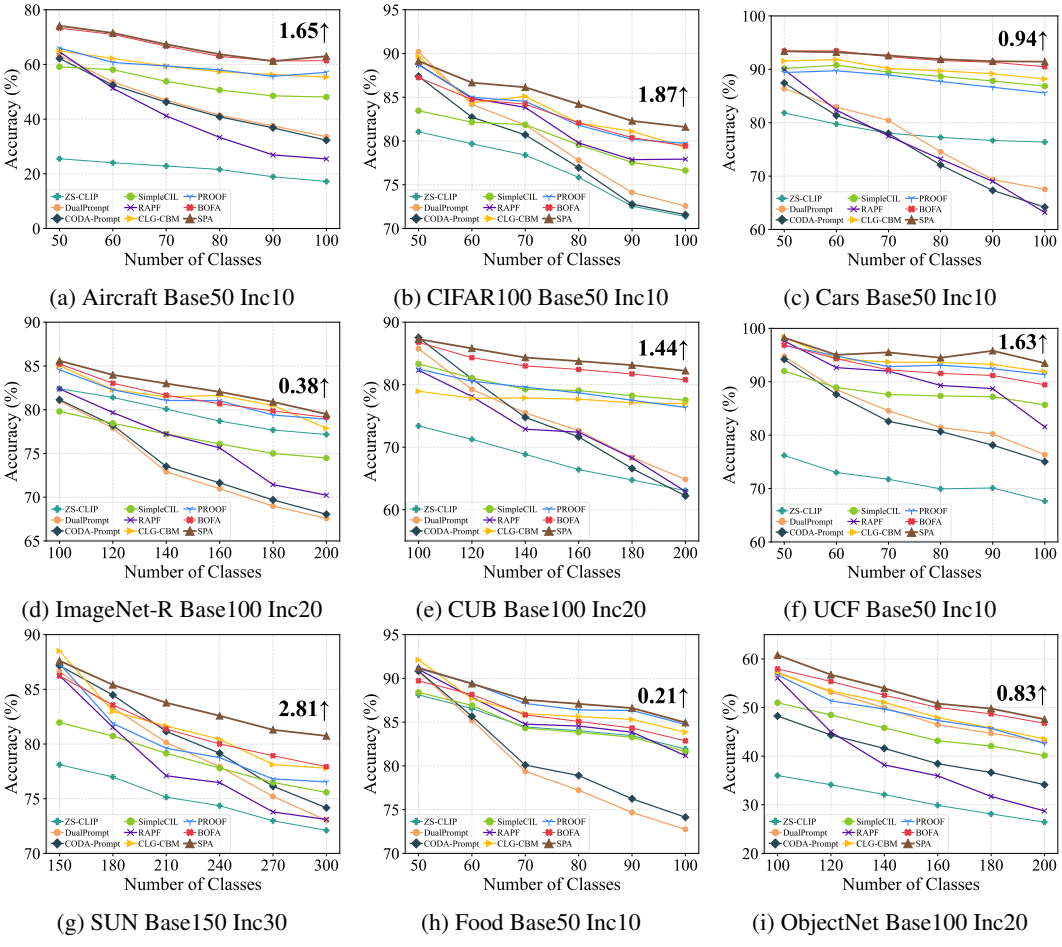


(a) Aircraft Base0 Inc10 (b) CIFAR100 Base0 Inc10 (c) Cars Base0 Inc10

(d) ImageNet-R Base0 Inc20 (e) CUB Base0 Inc20 (f) UCF Base0 Inc10

(g) SUN Base0 Inc30 (h) Food Base0 Inc10 (i) ObjectNet Base0 Inc20

Figure 10: Incremental performance of different methods on the B0 setting. We report the performance gap after the last incremental stage of SPA and the runner-up method at the end of the line.



(a) Aircraft Base50 Inc10 (b) CIFAR100 Base50 Inc10 (c) Cars Base50 Inc10

(d) ImageNet-R Base100 Inc20 (e) CUB Base100 Inc20 (f) UCF Base50 Inc10

(g) SUN Base150 Inc30 (h) Food Base50 Inc10 (i) ObjectNet Base100 Inc20

Figure 11: Incremental performance of different methods on a half-base setting. We report the performance gap after the last incremental stage of SPA and the runner-up method at the end of the line.