GATED MODALITY FUSION FOR CLIP-BASED CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

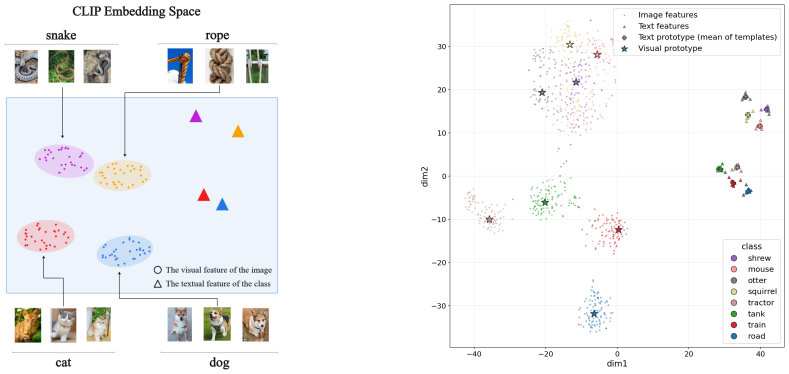
Class-Incremental Learning (CIL) enables models to continuously learn new classes without forgetting previously acquired knowledge. Due to the rich and transferable visual–text representations learned by Contrastive Language–Image Pre-training (CLIP) from large-scale image–text data, CIL methods employing CLIP as a pretrained backbone have recently achieved impressive performance. However, visual and textual representations provide complementary discriminative cues, whose relative effectiveness varies across class pairs. Most existing methods typically combine them using a fixed global ratio, implicitly assuming that their relative reliability is invariant across inputs and classes. We empirically show that this assumption is suboptimal: the optimal fusion ratio varies across class pairs and may change as the representation space evolves during incremental learning. In this paper, we propose GMF-CLIP, a Gated Modality Fusion framework that adaptively integrates CLIP's visual prototypes and textual classifiers through a learnable gating mechanism. For each input, GMF-CLIP dynamically estimates the modality contribution for each candidate class, enabling sample- and class-conditioned fusion. To sustain reliable gated fusion as the visual representation space evolves, we use the Visual Prototype Refresh (VPR) mechanism tailored to the gating network, which leverages changes in the learned fusion ratios to estimate representation shifts and calibrate old-class prototypes. Together, the gated fusion and prototype refresh mechanisms improve classifier reliability and reduce interference between old and new classes. Experimental results on four commonly used datasets demonstrate that GMF-CLIP achieves state-of-the-art performance.

1 INTRODUCTION

Class-Incremental Learning (CIL) aims to (Thrun, 1995; De Lange et al., 2021; Masana et al., 2022; Van de Ven & Tolias, 2019; Wang et al., 2024) continuously learn different tasks, yet models often suffer from the problem of catastrophic forgetting (French, 1999; McCloskey & Cohen, 1989; Ramasesh et al., 2021), i.e., they struggle to retain previously acquired knowledge while adapting to new tasks. Classical CIL models are typically trained from scratch (Thengane et al., 2022). However, this continuous training process can overwrite previously learned representations, substantially degrading performance on earlier classes.

Recently, pretrained vision–language models (Zhang et al., 2024; Zhou et al., 2022c), exemplified by Contrastive Language–Image Pre-training (CLIP) (Radford et al., 2021), have demonstrated remarkable zero-shot recognition capabilities, providing rich and transferable knowledge for CIL. Most existing CLIP-based CIL methods (Huang et al., 2024; Liu et al., 2025; Jha et al., 2024; D'Alessandro et al., 2023; Huang et al., 2025) typically preserve most pretrained knowledge by freezing the backbone and optimizing lightweight modules, such as prompts (Jia et al., 2022), low-rank adaptation (LoRA) modules (Hu et al., 2022), or adapters (Zhou et al., 2025b), for incremental adaptation. When exploiting both modalities, these methods often combine text-derived classifiers and image-derived visual prototypes using a fixed global ratio (Huang et al., 2024; Zhou et al., 2025b; He et al., 2026). Although effective, this strategy implicitly assumes that the relative discriminative strengths of the two modalities remain invariant across inputs and classes. However, different class pairs may favor different modalities for discrimination, making a single fixed fusion ratio unsuitable for all classes.

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.



(a) (b)

Figure 1: Modality-dependent class relationships in the embedding space of CLIP. (a) A schematic diagram. The visual features of cats and dogs are distinguishable but their textual features are similar, while the visual features of snakes and ropes are similar but their textual features are distinguishable. (b) t-SNE visualization of the CLIP embeddings of eight CIFAR-100 classes, including four visually similar animals (shrew, mouse, otter and squirrel) and four man-made classes (tractor, tank, train and road).

In practice, CLIP's image and text branches provide distinct yet complementary class information: image-derived prototypes capture task-specific visual characteristics, whereas text-derived classifiers preserve transferable semantic priors and inter-class relationships (Chen et al., 2025; Wolfe & Caliskan, 2022). Consequently, their relative discriminative strengths vary across class pairs. Visual prototypes may better distinguish semantically related classes with discriminative visual characteristics (e.g., "cat" and "dog"), whereas textual classifiers may better separate semantically unrelated classes with similar appearances (e.g., "snake" and "rope"). Figure 1(a) illustrates these modality-dependent class relationships within CLIP's shared embedding space. We further verify them on real data in Figure 1(b), where we visualize the image and text embeddings of eight CIFAR-100 classes produced by pre-trained CLIP. As new classes are introduced, the competition among observed classes continually changes: a new class may become a close visual or semantic neighbor of a previously learned class, while updates to the image encoder may further shift the geometry of visual prototypes. Therefore, the appropriate modality contribution should be dynamically reassessed for each input and candidate class throughout incremental learning. Methods using a fixed fusion ratio cannot capture these evolving and class-dependent differences in modality reliability.

In this paper, we first compare a fixed global fusion ratio with class-specific fusion weights learned by a lightweight multilayer perceptron. The results reveal that using a unified fusion ratio across all classes is suboptimal. Based on this observation, we propose GMF-CLIP, a gated modality fusion framework for CLIP-based CIL. GMF-CLIP employs a gating network to dynamically estimate the contributions of visual prototypes and textual classifiers for each input and candidate class. Furthermore, we introduce a Visual Prototype Refresh mechanism that estimates shifts in modality preference caused by updates to the image encoder and calibrates old-class visual prototypes.

Our main contributions are summarized as follows:

- We identify an overlooked limitation of existing modality-fusion strategies in CLIP-based CIL. Through empirical analysis, we show that using a fixed global fusion ratio is suboptimal because the relative discriminative strengths of visual prototypes and textual classifiers vary across inputs and candidate classes.
- We propose Gated Modality Fusion (GMF-CLIP), a new CLIP-based CIL framework that employs a gating network to dynamically estimate the contributions of visual prototypes and textual classifiers for each input and candidate class. We introduce a Visual Prototype Refresh (VPR) mechanism tailored to the gating network.
- We provide theoretical analyses of linear modality fusion and gated fusion, establishing the conditions under which modality fusion improves classification and adaptive gating outperforms a fixed fusion strategy.

- Extensive experiments on four standard CIL benchmarks demonstrate that GMF-CLIP achieves state-of-the-art performance among CLIP-based CIL methods, while ablation studies verify the contribution of each component.

The remainder of this paper is organized as follows. We first introduce related work. Then we present the problem setting of CIL and CLIP-based CIL and we introduce two parts of GMF-CLIP: Gating Network and Visual Prototype Refresh. Finally, we show the experiments and conclusions, respectively. The theoretical support for GMF-CLIP, experimental details and supplementary experiments are shown in Appendix.

2 Related Work

2.1 Class-Incremental Learning

Class-incremental learning (CIL) requires a model to learn disjoint sets of classes sequentially and, at inference time, distinguish among all classes observed so far without access to task identities. Traditional CIL approaches can be broadly grouped into replay-based, regularization-based, and architecture-based methods (Van de Ven & Tolias, 2019).

Replay-based methods either retain representative samples from previous tasks in an episodic memory (Rebuffi et al., 2017; Rolnick et al., 2019) or generate pseudo-samples that approximate past data distributions (Shin et al., 2017). Regularization-based methods (Kirkpatrick et al., 2017; Li & Hoiem, 2017) preserve previous knowledge by constraining changes to important parameters, representations, or predictions. Architecture-based methods (Ke et al., 2021; Rusu et al., 2016) allocate separate parameters or subnetworks to different tasks to reduce interference.

2.2 Pre-Trained Model-Based Class-Incremental Learning

The emergence of Vision Transformers (ViTs) has promoted parameter-efficient fine-tuning (PEFT) (Han et al., 2024) for CIL. Prompt-based CIL methods (Jia et al., 2022; Wang et al., 2022b;a; Smith et al., 2023) insert a small number of learnable parameters into a frozen backbone, while adapter-based methods (Chen et al., 2022; Yu et al., 2024) introduce compact task-specific modules. Low-rank adaptation methods (Hu et al., 2022; Liang & Li, 2024) instead represent parameter updates within low-dimensional subspaces. By optimizing only a small fraction of the model parameters, these approaches adapt PTMs to emerging classes while retaining their pre-trained representations.

The success of vision-language models, particularly Contrastive Language–Image Pre-training (CLIP), has provided another direction for PTM-based CIL. CLIP provides two complementary information within a shared embedding space: textual classifiers constructed from class names preserve transferable semantic relationships, whereas visual representations capture task-specific characteristics of observed images. Existing CLIP-based CIL methods exploit these multimodal priors from different perspectives. PROOF (Zhou et al., 2025b) freezes the image and text encoders and learns task-specific projection modules to integrate multimodal knowledge. RAPF (Huang et al., 2024) uses textual similarities to identify classes affected by incremental learning and adaptively adjusts their representations. MG-CLIP (Huang et al., 2025) mitigates forgetting by preserving and compensating for the image–text modality gap. These methods primarily employ textual information as classifiers, semantic anchors, regularization signals, or prototype-calibration priors and demonstrate the value of CLIP's multimodal priors.

3 Problem Formulation

In this section, we introduce the setting of Class-Incremental Learning and CLIP-based Class-Incremental Learning.

3.1 Problem Setting of Class-Incremental Learning

In the class-incremental learning setting, the model sequentially learns knowledge from a sequence of tasks $T_1, T_2, \cdots, T_K$. The training dataset of T_t can be represented as $\mathcal{D}_t = \{(x_n^t, y_n^t)\}_{n=1}^{N_t}$.

Here, N_t is the number of samples in training set $\mathcal{D}_t$ and $x_n^t \in \mathcal{X}_t$ belongs to the class $y_n^t \in \mathcal{Y}_t$, where $\mathcal{X}_t$ and $\mathcal{Y}_t$ represent the input space and the label space of task T_t. $\mathcal{Y}_i$ and $\mathcal{Y}_j$ are non-overlapping for $i \neq j$. Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cdots \cup \mathcal{X}_K$ and $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cdots \cup \mathcal{Y}_K$ denote the total input space and total output space, respectively. The aim is to train a feature extractor $f_\theta(\cdot)$ and a classifier $h_\phi(\cdot)$ to represent the function $\mathcal{F}$: $\mathcal{X} \to \mathcal{Y}$. During training task T_t, data in $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_{t-1}$ cannot be fully obtained. We use $\mathcal{M} = \{(\overline{x_i}, \overline{y_i})\}_{i=1}^m$ to denote the replay dataset, where m is the number of replay samples. The optimization objective can be expressed as follows:

$$(\theta^*, \phi^*) = \arg\min_{\theta,\phi} \mathbb{E}_{(x,y)\in\mathcal{D}_t\cup\mathcal{M}} \mathcal{L}_{CE}(h_\phi(f_\theta(x)), y), \tag{1}$$

where $\mathcal{L}_{CE}(\cdot)$ is the cross entropy loss.

3.2 CLIP-based Class-Incremental Learning

In our work, we utilize the CIL framework with CLIP as the pre-trained model. CLIP consists of a visual encoder g_v and a text encoder g_t. For an image sample $\mathbf{I}$, the visual embedding can be represented as $e_v = g_v(\mathbf{I})$. And for a class c, we construct a class-specific prompt p_c, such as "**a photo of a [CLASS]**$_c$" , and use $e_t^c = g_t(p_c)$ to represent the text feature of class c. The image $\mathbf{I}$ is classified by comparing the cosine similarity between e_v and e_t^c of all seen classes $\mathcal{Y}$:

$$P_c(\mathbf{I}) = \frac{\exp\left(\cos\left(e_v, e_t^c\right)/\tau\right)}{\sum_{q\in\mathcal{Y}} \exp\left(\cos\left(e_v, e_t^q\right)/\tau\right)}, \tag{2}$$

where τ is a temperature parameter. We typically use $P_c(\mathbf{I})$ for zero-shot classification.

Most existing CLIP-based CIL approaches combine visual prototypes and textual features as classifiers: $\overline{e}^c = \frac{\lambda p_v^c + (1-\lambda) e_t^c}{\|\lambda p_v^c + (1-\lambda) e_t^c\|}$. The classification is given as follows:

$$\overline{P_c(\mathbf{I})} = \frac{\exp\left(\cos\left(e_v, \overline{e}^c\right)/\tau\right)}{\sum_{q\in\mathcal{Y}} \exp\left(\cos\left(e_v, \overline{e}^q\right)/\tau\right)}. \tag{3}$$

where

$$p_v^c = \frac{\sum_{(x,y)\in\mathcal{D}} \mathbb{I}(y=c)\cdot g_v(x)}{\sum_{(x,y)\in\mathcal{D}} \mathbb{I}(y=c)} \tag{4}$$

is the visual prototype of class c and $\lambda \in [0, 1]$ is a fixed ratio.

In our work, we freeze the text encoder and only fine-tune the visual encoder during the incremental session. The text encoder provides relatively stable semantic anchors for class names, while the visual encoder needs to adapt to the evolving image distribution introduced by new classes. This design preserves the semantic structure of CLIP while allowing the visual representation to be updated for new tasks.

4 GMF-CLIP

In this section, we first empirically validate the limitations of fixed-ratio modality fusion in existing CLIP-based CIL methods. We then introduce our Gated Modality Fusion framework (GMF-CLIP).

4.1 Empirical Study of CLIP-based CIL Methods

As discussed in Introduction, the visual encoder captures fine-grained appearance information, while the text encoder encodes semantic relationships between classes. For different class pairs, the optimal modality for classification varies. To verify whether a uniform modality fusion ratio limits the classifier's performance, we compare the fixed modality fusion strategy and an adaptive modality fusion strategy.

We firstly use grid search to find the optimal fixed modality fusion ratio and use the linear combination of the visual prototype and the text feature as the classifier.

Then instead of a fixed ratio, we train a two-layer MLP to achieve an input-adaptive modality fusion ratio for each class. Figure 2 compares single-modality (visual-only, text-only), fixed-ratio fusion, and adaptive fusion on three benchmarks. The results show that adaptive modality fusion consistently outperforms fixed-ratio fusion, indicating that existing methods underutilize the complementary information in both modalities.

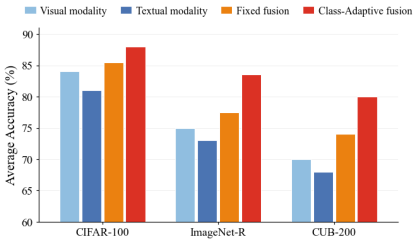


Figure 2: Average Accuracy (%) of different modality fusion methods.

The results demonstrate the benefit of the adaptive modality fusion. However, directly applying such a gating network to class-incremental learning does not preserve previously learned modality preferences and may produce unstable predictions under input perturbations. We therefore develop GMF-CLIP by introducing continual-learning constraints for the gating network and a modality-shift-guided prototype-refresh mechanism.

4.2 Gating Network

To address the problem raised above and maximize the utilization of pre-trained information contained within CLIP, we introduce a gating network to predict an adaptive modality fusion ratio for each input and candidate class.

Specifically, a gating network $G_t(\cdot)$ takes the image feature e_v as input and predicts a per-class fusion weight $\lambda \in [0, 1]$ that dynamically balances the contributions of the visual prototype and text feature during forward propagation. Thus, the model can adaptively determine the preferred modality for each class instead of relying on a fixed or hand-designed combination.

Formally, the gating network produces a fusion-weight vector $\boldsymbol{\lambda} = G_t(e_v) \in [0, 1]^{|\mathcal{Y}|}$, whose c-th element λ^c specifies the modality preference for class c. The fused classifier for class c is obtained by combining the visual prototype p_v^c and the text feature e_t^c according to the learned weight λ^c:

$$\bar{e}^c = F\left(\lambda^c p_v^c, (1-\lambda^c) e_t^c\right), \tag{5}$$

where F denotes the fusion operator. In this work, we instantiate F as linear interpolation. The final prediction is made by comparing the input visual feature e_v with the fused classifiers $\{\bar{e}^c\}_{c\in\mathcal{Y}}$. In this way, the gating network is incorporated into the forward propagation process and updated throughout the entire training process. As shown in Appendix A.3, the cross-entropy gradient encourages the gate to favor the modality better aligned with the input, thereby reflecting the relative reliability of the two modalities.

Overview of GMF-CLIP. Figure 3 shows the whole framework. For an input image $\mathbf{I}$, we obtain its visual embedding $e_v = g_v(\mathbf{I})$ from the image encoder, and the text features $\{e_t^c\}$ of all seen classes from the frozen text encoder. The visual embeddings are also used to maintain a visual prototype p_v^c for each class. The gating network takes e_v as input and outputs a fusion weight λ^c for every candidate class, and the classifier of class c is obtained by combining p_v^c and e_t^c according to λ^c. The prediction is made by comparing e_v with the fused classifiers $\{\bar{e}^c\}$. Several designs are introduced for the incremental setting. First, the output layer of the gating network grows with the label space: when task T_t arrives, the units of the old classes are kept and new units are appended for $\mathcal{Y}_t$ using Xavier uniform initialization, and the two gating-related losses keep the modality preference stable. Second, the visual prototypes of the old classes are refreshed every k epochs, where the change of the average gating value Δ_c decides the refresh intensity of each class. We describe these two parts in the following two subsections.

The Strategy to Prevent Forgetting for the Gating Network. We incorporate the gating network into the class-incremental learning framework. During the training phase, the similarity relationships between classes change as new tasks are introduced. Therefore, the gating network needs to be updated during the incremental session. To retain the modality-selection knowledge encoded by the previous gating network, we introduce a gate-retention loss on replay samples.

$$\mathcal{L}_{GN} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{I}\in\mathcal{M}} \|G_t(g_v^t(\mathbf{I}))_{\mathcal{Y}_{1:t-1}} - G_{t-1}(g_v^{t-1}(\mathbf{I}))\|_2, \tag{6}$$

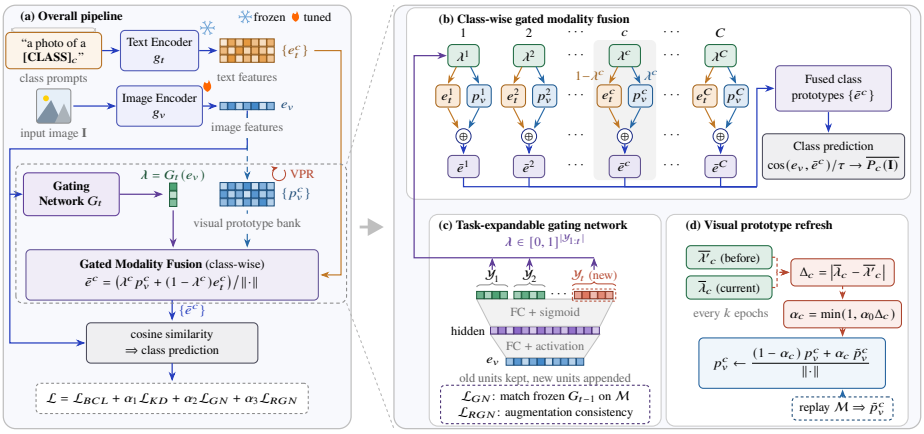


Figure 3: Overall framework of GMF-CLIP. (a) Pipeline: the frozen text encoder and tuned image encoder extract class text features and visual embeddings; the gating network predicts class-wise fusion weights to form fused classifiers for prediction. (b) Per-class adaptive fusion of the visual prototype and text feature with weight λ^c. (c) Gating network expansion for new tasks, with $\mathcal{L}_{GN}$ and $\mathcal{L}_{RGN}$ preserving the stability of modality preferences. (d) Gating-shift-guided refresh of old-class visual prototypes toward replay-based estimates.

where $|\mathcal{M}|$ is the number of replay samples. G_t denotes the current trainable gating network, whereas G_{t-1} is a frozen copy of the gating network obtained before learning task T_t. By matching the fusion weights produced before and after learning the current task, this objective preserves the modality-selection behavior learned for previously seen classes.

Enhancing the Robustness of the Gating Network. Since the gating network takes visual features as input, its output may be sensitive to image-level perturbations or geometric transformations. Such sensitivity can lead to unstable modality selection, especially in CIL scenarios. To encourage the gating network to learn augmentation-invariant modality reliability, we introduce the following augmentation-consistency loss:

$$\mathcal{L}_{RGN} = \mathbb{E}_{\mathbf{I}\sim\widetilde{\mathcal{D}}_t}\left[\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\left\|G_t\big(g_v^t(a(\mathbf{I}))\big) - G_t\big(g_v^t(\mathbf{I})\big)\right\|_2\right], \tag{7}$$

where $\widetilde{\mathcal{D}}_t = \mathcal{D}_t \cup \mathcal{M}$ denotes the entire training data during learning task T_t and $\mathcal{A}$ denotes a set of label-preserving augmentation transformations. This loss encourages the gating network to produce consistent modality-fusion ratios for the original and augmented views, thereby improving the robustness of the gating network.

4.3 Visual Prototype Refresh Based on Modality Shift

In the class-incremental learning scenario, the introduction of new classes alters the relative relationship of visual and textual modalities across the entire representation space, thereby influencing the optimal modality fusion ratio for each class. Since the gating network adaptively learns modality reliability, when the model learns a new task, the gating ratios for old classes may also change, leading to new modality preferences for the model in the updated representation space.

Therefore, we treat the change of gating ratios as a measure of "modality shift" and use it to control the intensity of the update to the prototypes of old classes. In our model, we update the visual prototype every k epochs during training. For old class c, we use the difference between the average gating value at the current stage $\overline{\lambda}_c$ and the average gating value before updating the visual prototypes $\overline{\lambda'}_c$.

$$\Delta_c = \left|\overline{\lambda}_c - \overline{\lambda'}_c\right|. \tag{8}$$

A large modality shift suggests that the relative reliability of the two modalities for class c has changed substantially, indicating a greater need to recalibrate its visual prototype in the updated

Table 1: Experiment results (%) on Split CIFAR-100 and Split ImageNet-R. R: replay (Y/N); Avg./Final: average/final accuracy. Bold/underlined means indicate best/second-best results.

Method	R	CIFAR-100		ImageNet-R (5 tasks)		ImageNet-R (10 tasks)	
		Avg.	Final	Avg.	Final	Avg.	Final
RAPF	N	86.14±0.28	78.04±0.34	82.05±0.31	71.12±0.39	81.26±0.35	70.48±0.42
CLG-CBM	N	84.49±0.26	76.82±0.37	82.83±0.29	77.61±0.33	82.17±0.32	76.34±0.36
CIL-PVLM	Y	82.50±0.41	71.90±0.45	84.89±0.27	74.65±0.38	84.46±0.30	73.87±0.40
CLAP4CLIP	Y	83.22±0.33	73.57±0.43	78.21±0.40	69.75±0.47	77.82±0.42	68.51±0.49
PROOF	Y	86.19±0.25	78.09±0.32	81.81±0.34	74.92±0.36	82.25±0.31	71.12±0.44
MG-CLIP+replay	Y	<u>86.47</u>±0.22	<u>80.20</u>±0.29	<u>85.43</u>±0.24	<u>79.35</u>±0.30	<u>85.42</u>±0.27	<u>78.59</u>±0.34
TPPT-V	Y	80.74±0.38	69.95±0.46	82.65±0.33	76.08±0.35	83.33±0.30	74.93±0.37
Ours	Y	**88.93**±0.18	**82.47**±0.24	**87.24**±0.20	**81.17**±0.26	**86.77**±0.23	**80.90**±0.28

Table 2: Experiment results(%) on Split ImageNet-R, Split CUB-200, and Split SUN397. R: replay (Y/N); Avg./Final: average/final accuracy. Bold/underlined means indicate best/second-best results.

Method	R	ImageNet-R (20 tasks)		CUB-200		SUN397	
		Avg.	Final	Avg.	Final	Avg.	Final
RAPF	N	81.17±0.36	70.03±0.44	79.09±0.41	62.77±0.52	79.59±0.38	67.87±0.48
CLG-CBM	N	81.56±0.34	71.00±0.43	<u>84.57</u>±0.26	76.48±0.35	81.09±0.32	72.56±0.39
CIL-PVLM	Y	82.65±0.31	74.20±0.38	80.43±0.37	69.75±0.45	81.02±0.35	72.37±0.41
CLAP4CLIP	Y	77.02±0.43	69.74±0.49	72.55±0.46	61.93±0.55	80.32±0.39	72.28±0.44
PROOF	Y	81.95±0.33	74.62±0.37	84.14±0.28	<u>76.53</u>±0.34	<u>83.85</u>±0.30	74.44±0.36
MG-CLIP+replay	Y	83.21±0.29	<u>76.95</u>±0.33	79.37±0.40	71.48±0.43	83.75±0.31	<u>77.23</u>±0.35
TPPT-V	Y	<u>83.59</u>±0.27	76.45±0.34	83.74±0.30	75.85±0.36	80.52±0.37	70.64±0.46
Ours	Y	**85.12**±0.21	**80.26**±0.27	**85.63**±0.22	**79.12**±0.29	**85.70**±0.24	**79.91**±0.28

representation space. Therefore we define the refresh intensity of class c as:

$$\alpha_c = \min(1, \alpha_0 \cdot \Delta_c), \tag{9}$$

where α_0 is the hyper-parameter that controls the refresh intensity. The gate-retention loss suppresses spurious changes in modality preference, whereas the residual changes that remain after retention reflect the functional adjustment required by the updated representation and class space. We use these residual changes as signals for prototype recalibration.

Then we compute a provisional prototype $\tilde{p}_v^c$ using replay samples and update the visual prototype as follows:

$$p_v^c \leftarrow \frac{(1-\alpha_c)\, p_v^c + \alpha_c \tilde{p}_v^c}{\| (1-\alpha_c)\, p_v^c + \alpha_c \tilde{p}_v^c \|}. \tag{10}$$

This mechanism does not rely on additional models. Instead, it utilizes the modality shift information provided by the gating network to adaptively adjust the update magnitude of the old class prototypes. This mechanism ensures that they maintain more consistent representations in the new representation space, thereby further enhancing the stability of the old classes.

4.4 Overall Optimization Objective

In this section, we will introduce the main training session of our work.

During the joint training session, we use a bidirectional contrastive loss $\mathcal{L}_{BCL}$ (Radford et al., 2021) as classification loss and a knowledge distillation loss $\mathcal{L}_{KD}$ (Hinton et al., 2015) to prevent forgetting.

To be more specific, given the visual feature e_v^i and the classifier $\bar{e}_i^c$ for class c, we define $s_{i,c} = (e_v^i)^\top \bar{e}_i^c/\tau$. The bidirectional contrastive loss is

$$\mathcal{L}_{\text{BCL}} = \frac{1}{2}\left[-\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \log \frac{\exp(s_{i,y_i})}{\sum_{c\in\mathcal{Y}_{1:t}}\exp(s_{i,c})} - \frac{1}{|\mathcal{C}_\mathcal{B}|}\sum_{c\in\mathcal{C}_\mathcal{B}} \log \frac{\sum_{i:y_i=c}\exp(s_{i,c})}{\sum_{i\in\mathcal{B}}\exp(s_{i,c})}\right], \tag{11}$$

where $\mathcal{C}_\mathcal{B}$ denotes the set of classes appearing in mini-batch $\mathcal{B}$ and τ is the scale parameter of CLIP.

Then we combine them with the forgetting and robustness losses of the gating network to form the overall optimization objective:

$$\mathcal{L} = \mathcal{L}_{BCL} + \alpha_1\mathcal{L}_{KD} + \alpha_2\mathcal{L}_{GN} + \alpha_3\mathcal{L}_{RGN}, \tag{12}$$

where $\alpha_1, \alpha_2, \alpha_3$ are the hyper-parameters that balance different components.

We conduct theoretical and empirical analyses to support GMF-CLIP, including hyperspherical fusion interpretation, margin-based analysis, gating-weight dynamics, and image-blur verification of modality reliability. Details are provided in Appendix.

5 Experiments

5.1 Experimental Setup

We evaluate GMF-CLIP on four datasets: Split CIFAR-100, Split ImageNet-R, Split CUB-200 and Split SUN397. We provide detailed descriptions of these datasets in Appendix.

We compare our GMF-CLIP with five replay-based CLIP-based CIL methods CIL-PVLM (Liu et al., 2023), CLAP4CLIP (Jha et al., 2024), MG-CLIP (Huang et al., 2025), TPPT (Lu et al., 2025) and PROOF (Zhou et al., 2025b) and two exemplar-free CLIP-based CIL methods RAPF (Huang et al., 2024) and CLG-CBM (Yu et al., 2025).

We evaluate each method on four datasets and compare them using the following two CIL metrics: Final Accuracy and Average Accuracy.

Final Accuracy is the average classification accuracy of N tasks after the whole training process, i.e. $AA_N = \frac{1}{N}\sum_{i=1}^{N} A_{i,N}$, where $A_{i,N}$ is the average accuracy of all classes in task i after training task N.

Average Accuracy is the average of the classification accuracy after each incremental session. The average accuracy of the model is expressed as $\overline{AA_N} = \frac{1}{N}\sum_{t=1}^{N} AA_t$, where the definition of AA_t is similar to the definition of AA_N.

In our experiment setup, we consider CLIP with ViT-B/16 pre-trained on LAION-400M as the pre-trained model for all CLIP-based CIL methods. We use Adam as the optimizer for GMF-CLIP. We set $m = 1000$ for all replay-based methods and follow (Rebuffi et al., 2017) to update the replay datasets. We repeat each method 3 times and report the average result in our paper. The details of the experiment setting and hyper-parameters are shown in Appendix. Our code is based on PyCIL (Zhou et al., 2023).

5.2 Main Results

The experimental results of GMF-CLIP and other PTM-based CIL methods are shown in Table 1 and Table 2. The final accuracy of GMF-CLIP outperforms other CLIP-based CIL methods by at least 1.82% and the average accuracy of GMF-CLIP outperforms other methods by at least 1.06%. The results show that GMF-CLIP has achieved the state-of-the-art performance among CLIP-based CIL approaches.

We further observe that the advantage of GMF-CLIP is more pronounced in challenging continual learning scenarios. For example, on Split ImageNet-R, GMF-CLIP improves the final accuracy over the strongest baseline by 2.31% in the 10-task setting and 3.31% in the 20-task setting. Similar improvements can also be observed on datasets such as Split CUB-200 and Split SUN397. These results indicate that GMF-CLIP constructs more reliable class representations and better preserves

old-class knowledge during incremental learning. We believe that this benefit mainly comes from the gating network, which adaptively balances visual and textual information and reduces the interference caused by new classes.

5.3 Ablation Study

To verify the effectiveness of each component of the model, we perform an ablation experiment on different datasets. We separately evaluated the average accuracy and final accuracy of GMF-CLIP after removing the gating network , removing two gating-related losses and replacing Visual Prototype Refresh (VPR) to a naive prototype update, which updates the visual prototype only according to the replay samples.

Table 3: Ablation results (%) of GMF-CLIP on Split CIFAR-100 and Split ImageNet-R (10 tasks).

Gating network	VPR	$\mathcal{L}_{\text{GN}} + \mathcal{L}_{\text{RGN}}$	CIFAR-100		ImageNet-R (10 tasks)	
			Avg.	Final	Avg.	Final
✗	✗	✗	85.61	76.93	82.26	73.48
✓	✗	✗	86.27	78.06	83.82	75.96
✓	✗	✓	86.75	79.11	85.24	77.00
✓	✓	✗	87.82	81.35	85.92	80.06
✓	✓	✓	**88.93**	**82.47**	**86.77**	**80.90**

The ablation results are shown in Table 3. Adding the gating network consistently improves both average and final accuracy over the baseline, showing the benefit of adaptive modality fusion. VPR and two gating-related loss also confirm their enhancements to the gating network and VPR plays a more important role in GMF-CLIP.

5.4 T-SNE Analysis of Gating Network

To verify that the gating network achieved our initial goal: adaptively fusing the two modalities, we visualize the embedding space with t-SNE on four classes from Split CIFAR-100 after full training.

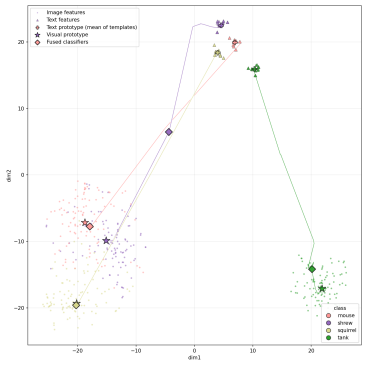


Figure 4: T-SNE visualization of GMF-CLIP.

Figure 4 shows that the gating network assigns different fusion strategies across classes. Among the three rodent classes with mixed image features, shrew leans most toward the text prototype, while tank, with a compact visual cluster, keeps its fused classifier close to the visual prototype. This qualitatively indicates that the gating network can capture class-specific modality preferences, consistent with Figure 1.

More experimental and efficiency analysis are shown in Appendix.

6 Conclusion

In this work, we analyzed the sub-optimality problem of existing CLIP-based CIL methods experimentally. Then, we proposed GMF-CLIP, a novel input-adaptive CLIP-based CIL method that employs a gating network to control the ratio of linear modality fusion for each class. Furthermore, we proposed a Visual Prototype Refresh Based on Modality Shift (VPR) to calibrate the prototypes of old classes during the incremental phase. Differing from other CLIP-based CIL methods, we introduce a query-adaptive, class-wise classifier fusion mechanism that dynamically balances stable textual knowledge and evolving visual knowledge in CIL. Experimental results on four datasets confirmed that our approach achieved state-of-the-art performance. As GMF-CLIP operates at the modality-fusion level, it is complementary to some PEFT-based incremental learning methods. This makes combining GMF-CLIP with such methods worth exploring in future work.

AI USE STATEMENT

In this work, generative AI tools were used to assist with writing parts of the code, literature searches, and manuscript drafting and language editing. The authors carefully review AI-assisted content and take full responsibility for the content and accuracy of the entire paper.

REFERENCES

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

Yiyang Chen, Tianyu Ding, Lei Wang, Jing Huo, Yang Gao, and Wenbin Li. Enhancing few-shot class-incremental learning via training-free bi-level modality calibration. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9881–9890, 2025. doi: 10.1109/CVPR52734.2025.00923.

Marco D'Alessandro, Alberto Alonso, Enrique Calabrés, and Mikel Galar. Multimodal parameter-efficient few-shot class incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3393–3403, 2023.

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.

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

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Lingfeng He, De Cheng, Di Xu, Huaijie Wang, and Nannan Wang. Harnessing textual semantic priors for knowledge transfer and refinement in clip-driven continual learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(26):21645–21653, 2026. doi: 10.1609/aaai.v40i26.39314.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

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.

Saurav Jha, Dong Gong, and Lina Yao. Clap4clip: Continual learning with probabilistic finetuning for vision-language models. *Advances in neural information processing systems*, 37:129146–129186, 2024.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34: 22443–22456, 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.

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

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024.

Wenzhuo Liu, Fei Zhu, Longhui Wei, and Qi Tian. C-clip: Multimodal continual learning for vision-language model. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xialei Liu, Xusheng Cao, Haori Lu, Jia-wen Xiao, Andrew D Bagdanov, and Ming-Ming Cheng. Class incremental learning with pre-trained vision-language models. *arXiv preprint arXiv:2310.20348*, 2023.

Haodong Lu, Xinyu Zhang, Kristen Moore, Jason Xue, Lina Yao, Anton van den Hengel, and Dong Gong. Continual learning on CLIP via incremental prompt tuning with intrinsic textual anchors. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=YJnjkzKq5Y.

Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2000. ISBN 9780471953333.

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.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

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.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International conference on learning representations*, 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.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Abhishek Kumar Sinha, Nitant Dube, and Soma Biswas. Quantized residuals to continuous prompts for few-shot class incremental learning in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3856–3865, 2026.

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.

Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022.

Sebastian Thrun. A lifelong learning perspective for mobile robot control. In *Intelligent robots and systems*, pp. 201–214. Elsevier, 1995.

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

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, 2022a.

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, 2022b.

Robert Wolfe and Aylin Caliskan. Contrastive visual semantic pretraining magnifies the semantics of natural language representations. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3050–3061, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.217. URL https://aclanthology.org/2022.acl-long.217/.

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. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14976–14986, 2025. URL https://api.semanticscholar.org/CorpusID:277451998.

Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12455–12464, 2021.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9046–9056, 2022a.

Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12816–12831, 2022b.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: a python toolbox for class-incremental learning. *SCIENCE CHINA Information Sciences*, 66(9):197101, 2023. doi: https://doi.org/10.1007/s11432-022-3600-y.

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, 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*, 2025b.

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, 2022c.

Yixiong Zou, Shanghang Zhang, Yuhua Li, and Ruixuan Li. Margin-based few-shot class-incremental learning with class-level overfitting mitigation. *Advances in neural information processing systems*, 35:27267–27279, 2022.

APPENDIX

This appendix provides additional theoretical and empirical analyses supporting GMF-CLIP. Appendix A first presents a hyperspherical product-of-experts interpretation under the von Mises–Fisher distribution (Mardia & Jupp, 2000), which explains why linear visual–textual fusion is compatible with normalized CLIP embeddings. We then provide a margin-based view showing why a fixed unified fusion ratio is structurally limited and why adaptive fusion weights are needed. Next, we analyze the optimization dynamics of the gating weight and show that the learned gate is encouraged to reflect the relative reliability of visual and textual cues. Finally, we conduct a controlled image-blur experiment to empirically verify that the learned gate responds to reduced visual reliability. Appendix B shows the descriptions of used datasets and the detailed experimental settings. Appendix C shows the performance of GMF-CLIP in a Few-Shot Class-Incremental Learning (FSCIL) scenario, which demonstrates that our method is equally applicable to more realistic and complex scenarios. Finally, we analyze the memory and training efficiency in Appendix D.

A ADDITIONAL THEORETICAL AND EMPIRICAL ANALYSIS

A.1 PRODUCT-OF-EXPERTS VIEW UNDER VMF

A.1.1 THE VMF DISTRIBUTION

To provide a geometric interpretation of linear visual–text fusion on normalized embeddings, we adopt a product-of-experts (PoE) perspective based on the von Mises–Fisher (vMF) distribution: on the hypersphere, the product of two experts remains a vMF distribution. This perspective explains *why linear modality fusion is geometrically compatible with normalized CLIP embeddings*.

The von Mises–Fisher distribution can be regarded as the hyperspherical analogue of the Gaussian distribution:

$$p(\mathbf{x} \mid \boldsymbol{\mu}, \kappa) = C_d(\kappa) \exp(\kappa\, \boldsymbol{\mu}^\top \mathbf{x}).$$

Here, $\boldsymbol{\mu} \in \mathbb{S}^{d-1}$ denotes the mean direction, $\kappa \geq 0$ is the concentration parameter, and $C_d(\kappa)$ the normalization constant. We abbreviate this distribution as $\mathrm{vMF}(\boldsymbol{\mu}, \kappa)$.

To connect the formulation with the PoE interpretation, let the visual and text embeddings of class c be characterized by directions $\hat{\boldsymbol{\mu}}_v^c$ and $\hat{\boldsymbol{\mu}}_t^c$, respectively (in CLIP, the outputs of g_v and g_t are typically L_2-normalized). If the visual direction is estimated from the training samples $\{e_v^{(i)}\}$ of class c (unit vectors), we use the sample mean direction

$$\hat{\boldsymbol{\mu}}_v^c = \frac{\bar{e}_v^c}{\|\bar{e}_v^c\|}, \quad \bar{e}_v^c = \frac{1}{N_c} \sum_i e_v^{(i)}.$$

The resultant mean length $\|\bar{e}_v^c\|$ can further define the classical vMF estimate of the concentration parameter, $\hat{\kappa}_v^c = f_d^{-1}(\|\bar{e}_v^c\|)$, where f_d is a monotone mapping related to the Bessel function. On the text side, the prompt embedding $e_t^c = g_t(p_c)$ is normalized and used as the direction, i.e., $\hat{\boldsymbol{\mu}}_t^c = e_t^c / \|e_t^c\|$.

A.1.2 VMF PRODUCT-OF-EXPERTS MODEL

We now state the basic closure property that motivates our PoE interpretation.

Proposition 1. The product of two vMF distributions (up to a normalization factor) is still a vMF distribution:

$$\begin{aligned} &\mathrm{vMF}(\boldsymbol{\mu}_1, \kappa_1) \otimes \mathrm{vMF}(\boldsymbol{\mu}_2, \kappa_2) \propto \mathrm{vMF}(\boldsymbol{\mu}_f, \kappa_f), \\ &\kappa_f \boldsymbol{\mu}_f = \kappa_1 \boldsymbol{\mu}_1 + \kappa_2 \boldsymbol{\mu}_2. \end{aligned} \tag{13}$$

Proof. Using the vMF density form,

$$p_i(\mathbf{x}) \propto \exp(\kappa_i \boldsymbol{\mu}_i^\top \mathbf{x}), \quad i \in \{1, 2\},$$

their product satisfies

$$p_1(\mathbf{x}) p_2(\mathbf{x}) \propto \exp\Big((\kappa_1 \boldsymbol{\mu}_1 + \kappa_2 \boldsymbol{\mu}_2)^\top \mathbf{x}\Big).$$

Let $\boldsymbol{\eta}_f = \kappa_1\boldsymbol{\mu}_1 + \kappa_2\boldsymbol{\mu}_2$. Writing $\kappa_f = \|\boldsymbol{\eta}_f\|$ and $\boldsymbol{\mu}_f = \boldsymbol{\eta}_f/\|\boldsymbol{\eta}_f\|$, we obtain

$$p_1(\mathbf{x})p_2(\mathbf{x}) \propto \exp(\kappa_f \boldsymbol{\mu}_f^\top \mathbf{x}),$$

which is again a vMF density after reintroducing the normalization constant. □

Therefore, for a sample $\mathbf{x}$, the log-density of the fused vMF is

$$\begin{aligned} \log p(\mathbf{x}) &= \kappa_f \boldsymbol{\mu}_f^\top \mathbf{x} + \log C_d(\kappa_f) \\ &= (\kappa_1\boldsymbol{\mu}_1 + \kappa_2\boldsymbol{\mu}_2)^\top \mathbf{x} + \log C_d(\kappa_f). \end{aligned} \tag{14}$$

Under the CLIP setting, however, classification is performed by cosine similarity on normalized embeddings. Therefore, after forming the fused natural parameter, we retain only its direction and compare it with the normalized query. For class c, let

$$\boldsymbol{\eta}_f^c = \lambda^c \hat{\boldsymbol{\mu}}_v^c + (1-\lambda^c)\hat{\boldsymbol{\mu}}_t^c, \qquad \hat{\boldsymbol{\mu}}_f^c = \frac{\boldsymbol{\eta}_f^c}{\|\boldsymbol{\eta}_f^c\|},$$

where $\lambda^c \in [0,1]$ is the class-c fusion coefficient predicted by the gating network. The score for class c is thus written as

$$\text{score}_c = (\hat{\boldsymbol{\mu}}_f^c)^\top \hat{e}_v = \frac{\left(\lambda^c\,\hat{\boldsymbol{\mu}}_v^c + (1-\lambda^c)\,\hat{\boldsymbol{\mu}}_t^c\right)^\top \hat{e}_v}{\left\|\lambda^c\,\hat{\boldsymbol{\mu}}_v^c + (1-\lambda^c)\,\hat{\boldsymbol{\mu}}_t^c\right\|},$$

which is exactly the cosine similarity between the normalized fused direction and the normalized query. From the PoE viewpoint, the numerator is induced by the addition of the two modal experts in the natural-parameter space, while the denominator appears because CLIP keeps only the fused direction and discards the concentration magnitude in the final classifier. If one further adopts a shared-concentration approximation across classes, then the additive term $\log C_d(\kappa)$ becomes irrelevant to the $\arg\max$ decision. Therefore, under hyperspherical embeddings, gated fusion weighted by λ^c and $1-\lambda^c$ can be interpreted as first combining two modal vMF experts in the natural-parameter space and then normalizing the result for cosine-based classification, while λ^c simultaneously acts as the relative confidence (effective κ) assigned to the visual and textual experts before normalization.

A.2 A Margin-based View of Fixed and Adaptive Fusion

We now analyze the limitation of a fixed fusion ratio from a pairwise-margin perspective. To provide a intuitive understanding, we use the unnormalized model for our analysis. For an input x with ground-truth class y and a competing class j, let

$$\Delta_{y,j}^v(x) = A_y(x) - A_j(x), \qquad \Delta_{y,j}^t(x) = B_y(x) - B_j(x),$$

where $A_c(x) = \hat{e}_v(x)^\top \hat{\mu}_v^c$ and $B_c(x) = \hat{e}_v(x)^\top \hat{\mu}_t^c$ denote the visual and textual alignments. Under a fixed unified coefficient λ, the pairwise margin is a convex combination of the two modality margins,

$$m_{y,j}(x;\lambda) = \frac{1}{\tau}\left[\lambda\Delta_{y,j}^v(x) + (1-\lambda)\Delta_{y,j}^t(x)\right],$$

with the same trade-off imposed on all samples and class pairs. As a reference point for adaptive fusion, consider an oracle gate that selects the larger margin for each triple (x,y,j):

$$m_{y,j}^{\text{oracle}}(x) = \frac{1}{\tau}\max\left\{\Delta_{y,j}^v(x), \Delta_{y,j}^t(x)\right\}.$$

Proposition 2. With the expectation taken over input samples and competing classes,

$$\mathbb{E}\left[m_{y,j}^{\text{oracle}}(x)\right] \geq \max_{\lambda\in[0,1]} \mathbb{E}\left[m_{y,j}(x;\lambda)\right],$$

and the inequality is strict when the dominant modality varies across samples or class pairs with nonzero probability.

Proof. $\mathbb{E}[m_{y,j}(x;\lambda)]$ is linear in λ, so the best fixed ratio attains $\frac{1}{\tau}\max\{\mathbb{E}\Delta^v_{y,j}, \mathbb{E}\Delta^t_{y,j}\}$, whereas the oracle attains $\frac{1}{\tau}\mathbb{E}[\max\{\Delta^v_{y,j}, \Delta^t_{y,j}\}]$. The claim follows from $\mathbb{E}[\max\{U,V\}] \geq \max\{\mathbb{E}[U], \mathbb{E}[V]\}$, which is strict unless one modality dominates almost surely. □

This does not claim that the learned gate attains the oracle margin; it identifies a structural limitation of fixed-ratio fusion, namely that a single coefficient can only optimize an average trade-off, while adaptive fusion can exploit sample- and class-dependent modality preferences. This is consistent with the gain of adaptive over fixed-ratio fusion observed in our preliminary study.

A.3 Optimization Dynamics of Modality Reliability Estimation

To elucidate the mechanism by which the gating network adaptively estimates image clarity through the fusion weight λ^c, we analyze the optimization dynamics under the cross-entropy objective. Let s_c denote the classification score for the ground-truth class c, and P_c denote the corresponding predicted probability. By considering the normalized fused classifier, we have

$$e^c = \frac{\lambda^c p_v^c + (1-\lambda^c)e_t^c}{\|\lambda^c p_v^c + (1-\lambda^c)e_t^c\|_2}.$$

Let

$$z_c = \lambda^c p_v^c + (1-\lambda^c)e_t^c, \qquad r_c = \|z_c\|_2.$$

Then the classification score for class c is

$$s_c = \frac{1}{\tau}\cos(e_v, e^c) = \frac{1}{\tau}\cos\left(e_v, \frac{z_c}{r_c}\right).$$

We define

$$A = \cos(e_v, p_v^c), \qquad B = \cos(e_v, e_t^c), \qquad C = \cos(p_v^c, e_t^c).$$

Since e_v, p_v^c, and e_t^c are L_2-normalized, the score can be written as

$$s_c = \frac{1}{\tau}\cdot\frac{\lambda^c A + (1-\lambda^c)B}{r_c},$$

where

$$r_c^2 = (\lambda^c)^2 + (1-\lambda^c)^2 + 2\lambda^c(1-\lambda^c)C.$$

Therefore, the partial derivative of the score with respect to the fusion weight is

$$\begin{aligned}\frac{\partial s_c}{\partial \lambda^c} &= \frac{1}{\tau r_c}\Bigg[(A-B) - \frac{\lambda^c A + (1-\lambda^c)B}{r_c^2} \\ &\qquad \times (2\lambda^c - 1)(1-C)\Bigg] \\ &\approx \frac{1}{\tau r_c}(A-B).\end{aligned} \tag{15}$$

The first term $A-B$ captures the visual–textual alignment gap, while the second term comes from the derivative of the normalization factor.

Under the cross-entropy loss $\mathcal{L}_{CE} = -\log\frac{\exp(s_c)}{\sum_j \exp(s_j)}$, the gradient with respect to s_c is derived as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{CE}}{\partial s_c} &= \frac{\partial}{\partial s_c}\left[-s_c + \log\sum_j \exp(s_j)\right] \\ &= -1 + \frac{\exp(s_c)}{\sum_j \exp(s_j)} = P_c - 1.\end{aligned}$$

Applying the chain rule, the gradient propagated to the gating weight λ^c is:

$$\frac{\partial \mathcal{L}_{CE}}{\partial \lambda^c} = \frac{\partial \mathcal{L}_{CE}}{\partial s_c}\frac{\partial s_c}{\partial \lambda^c} \approx (P_c - 1)\frac{1}{\tau r_c}(A-B).$$

This equation suggests that the update direction of λ^c is determined by the sign of the alignment gap $(A - B)$, given that $P_c - 1 \leq 0$ acts as a negative feedback signal. Specifically, for high-clarity images where $A > B$, the gradient drives λ^c upward to prioritize discriminative visual cues. Conversely, for noisy or blurred inputs where $A < B$, the optimization reduces λ^c to shunt the decision toward the more stable textual anchor.

This gradient-driven mechanism essentially forces the gating network to approximate the intrinsic precision field of the CLIP latent manifold. Since this precision field characterizes the geometric reliability of feature vectors rather than their semantic labels, the learned gating logic generalizes naturally to unseen categories in incremental tasks. Consequently, λ^c functions as a class-agnostic reliability estimator, enabling GMF-CLIP to achieve an optimal balance between visual grounding and semantic abstraction without over-relying on specific training samples.

A.4 Empirical Verification of Modality Reliability under Image Blur

To further support the optimization-based interpretation above, we conduct a controlled image-degradation experiment on the *final model* after the last incremental task. Specifically, after finishing the training of task 9, we freeze all model parameters, including the visual encoder, the gating network, and the class prototypes. We then evaluate this fixed model on the test set under different levels of Gaussian blur. Therefore, the only varying factor in this experiment is the blur strength of the input image, while the model itself remains unchanged.

For each test image I_i with ground-truth label y_i, we generate a sequence of blurred images

$$I_i^{(\sigma)} = \mathcal{B}_\sigma(I_i),$$

where $\mathcal{B}_\sigma(\cdot)$ denotes Gaussian blur with standard deviation σ. We use

$$\sigma \in \{0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\},$$

where $\sigma = 0$ corresponds to the original image.

For each blurred image $I_i^{(\sigma)}$, we record the gating weight assigned to the ground-truth class:

$$\lambda_{y_i}(I_i^{(\sigma)}).$$

We then average this quantity over all test samples:

$$\overline{\lambda}_y(\sigma) = \frac{1}{N} \sum_{i=1}^{N} \lambda_{y_i}(I_i^{(\sigma)}),$$

where N is the number of test samples.

In addition, we compute the visual-text alignment gap for the ground-truth class:

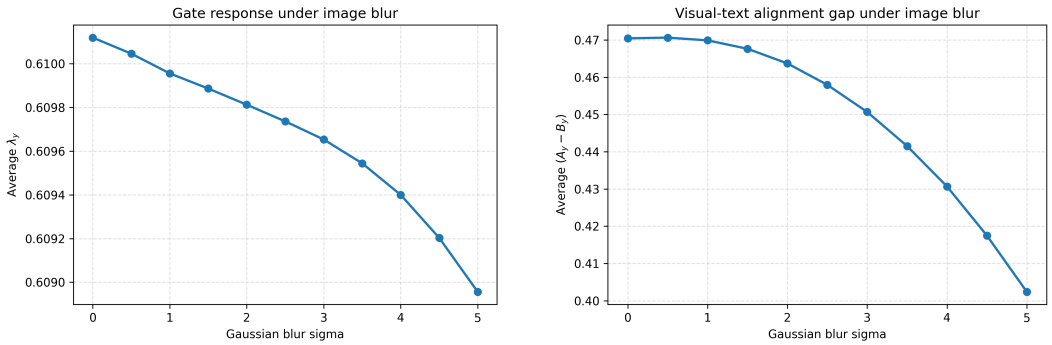
$$\Delta_y(I_i^{(\sigma)}) = A_{y_i}(I_i^{(\sigma)}) - B_{y_i}(I_i^{(\sigma)}),$$

where

$$A_{y_i}(I_i^{(\sigma)}) = \cos\left(e_v(I_i^{(\sigma)}), p_v^{y_i}\right),$$
$$B_{y_i}(I_i^{(\sigma)}) = \cos\left(e_v(I_i^{(\sigma)}), e_t^{y_i}\right).$$

This quantity directly corresponds to the alignment gap $(A - B)$ in the gradient analysis above.

Figure 5 reports the results on the final model after task 9. As the blur strength increases, the average visual gate $\overline{\lambda}_y(\sigma)$ shows a consistent decreasing trend. This indicates that the learned gate tends to assign a smaller weight to the visual prototype when the visual evidence becomes less reliable. Meanwhile, the visual-text alignment gap $\Delta_y(\sigma)$ also decreases under stronger blur, which is consistent with the theoretical stated upon: when the visual alignment becomes weaker relative to the textual alignment, the optimization dynamics favor a smaller visual gate.



(a) Average visual gate $\overline{\lambda}_y(\sigma)$ under blur.

(b) Visual-text alignment gap $\Delta_y(\sigma)$ under blur.

Figure 5: Empirical verification of modality reliability estimation on the final model after task 9.

B Additional Experimental Details

B.1 Introduction to the Datasets

Split CIFAR-100 is a widely used class-incremental learning benchmark. It contains 100 classes which are randomly split into 10 incremental tasks of disjoint classes and each class has 500 training samples and 100 testing samples.

Split ImageNet-R was first adapted for class-incremental learning by Dual-Prompt (Wang et al., 2022a) and is more relevant to real-life scenarios. Similarly, it contains 200 classes which are randomly split into 5, 10 or 20 incremental tasks of disjoint classes and each class has 120 training samples and 30 testing samples.

Split CUB-200 is a commonly used class-incremental learning benchmark constructed from the CUB-200 dataset, where its 200 fine-grained bird species are evenly divided into 10 incremental tasks, each containing 20 classes. CUB-200 contains 5,994 training samples and 5,794 test samples in total.

Split SUN397 is a scene recognition benchmark adapted for class-incremental learning. Following the commonly used setting, 300 classes are selected from the original 397 classes and randomly split into 10 incremental tasks with disjoint class sets, where each task contains 30 classes and each class has 50 training samples and 50 testing samples. (Zhou et al., 2025a)

Incremental protocol and evaluation. We follow the single-head class-incremental learning setting, in which the task identity is unavailable at inference. After task T_t, a model predicts among all classes observed up to T_t and is evaluated on their cumulative test set. Unless otherwise stated, one random class order is generated using NumPy with seed 1993 and is used for all runs under the corresponding setting. Avg.Acc. is the arithmetic mean of the cumulative top-1 accuracies measured after individual tasks, whereas Fin.Acc. is the cumulative top-1 accuracy after the last task.

B.2 Gating Network Architecture

The gate takes the ℓ_2-normalized 512-dimensional CLIP image embedding as input. It consists of a shared linear layer of size $512 \to 256$, followed by a LeakyReLU activation with negative slope 0.01, and two sigmoid output branches. The first branch produces a class-specific vector $p(\mathbf{z}) \in [0,1]^{1000}$, while the second produces a sample-specific scalar $g(\mathbf{z}) \in [0,1]$. For C_t observed classes, only the first C_t entries are used, and the final coefficient for class c is

$$\lambda_c(\mathbf{z}) = \mathrm{clip}_{[0,1]}(p_c(\mathbf{z}) + g(\mathbf{z}) - 0.5)\,. \tag{16}$$

The coefficients are then used for the visual–text prototype fusion described in the main paper. All linear weights are initialized with Xavier uniform initialization and all biases are initialized to zero. The gate contains 388,585 trainable parameters. Its output capacity of 1000 classes is an implementation upper bound and is unrelated to the replay-memory budget.

Table 4: Results on CIFAR-100 under the standard 5-way 5-shot FSCIL protocol. Avg. denotes the mean accuracy over sessions 0–8. Results of prior methods are taken from the corresponding papers.

Method	**Accuracy in each session (%)**									**Avg.** ↑
	0	1	2	3	4	5	6	7	8	
CLOM (Zou et al., 2022)	74.20	69.83	66.17	62.39	59.26	56.48	54.36	52.16	50.25	60.57
PC (Zhang et al., 2021)	76.30	71.89	67.70	63.40	60.21	57.31	55.01	52.79	50.74	61.71
LIMIT (Zhou et al., 2022b)	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	61.84
FACT (Zhou et al., 2022a)	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	62.24
CLIP zero-shot	74.45	72.83	72.11	70.25	69.71	69.55	69.52	68.78	68.04	70.58
BiMC (Chen et al., 2025)	81.98	79.74	78.84	77.00	76.11	75.68	75.33	74.61	73.18	76.94
QR-Prompt (Sinha et al., 2026)	83.55	82.03	81.12	78.83	78.62	78.08	78.52	77.52	75.67	79.32
GMF-CLIP	**86.90**	**85.95**	**85.71**	**84.00**	**82.86**	**81.71**	**80.33**	**79.73**	**78.86**	**82.89**

B.3 Training and Implementation Details

In our experiment setup, we consider CLIP with ViT-B/16 pre-trained on LAION-400M as the pre-trained model for all CLIP-based CIL methods. The text encoder and logit-scale parameter remain frozen, whereas the image encoder and the gate are optimized. We use Adam ($\beta_1 = 0.9$, $\beta_2 = 0.9$) with a learning rate of 0.0001 as the optimizer for GMF-CLIP. We set $m = 1000$ for all replay-based methods and follow (Rebuffi et al., 2017) to update the replay datasets. We repeat each experiment three times and set the seed = 0,1,1993. The temperature of logit scale is the same as the original pre-trained CLIP. For the augmentation-consistency loss, we instantiate the augmentation set $\mathcal{A}$ using non-identity rotation transformations, i.e., $\mathcal{A} = \{r_{90^\circ}, r_{180^\circ}, r_{270^\circ}\}$. All input images are resized to a resolution of 224×224, normalized to $[0, 1]$, and trained with a batch size of 128. The number of training epochs is 10 for all datasets and we set the hyper-parameters $k = 3, \alpha_0 = 10, \alpha_1 = \alpha_2 = \alpha_3 = 1$ for all settings. We use logit distillation as the distillation loss and the distillation temperature is 2 in our experiments. We conduct these experiments on one NVIDIA GeForce RTX 4070 GPU. Our code is based on modifications to PyCIL (Zhou et al., 2023).

B.4 The Setting of Experments in Figure 2

For the fixed-ratio fusion ablation, we uniformly evaluate $\lambda \in \{0.0, 0.1, \ldots, 1.0\}$ and report the complete sweep. This grid is used only to analyze fixed visual–text fusion and is not used by GMF-CLIP, which predicts a sample- and class-dependent coefficient. The gate-specific settings are fixed to a hidden dimension of 256, a LeakyReLU slope of 0.01, 10/3 warm-up epochs for the first/subsequent tasks, and a prototype-refresh interval of 3. Common optimization, replay, and loss settings are reported in the main experimental setup.

C Experiments on Few-Shot Class-Incremental Learning

We additionally evaluate GMF-CLIP under the standard CIFAR-100 Few-Shot Class Incremental Learning (FSCIL) scenario. The first 60 classes form the base session, while the remaining 40 classes are introduced through eight 5-way 5-shot incremental sessions without replay. In this scenario, we did not use replay samples, we directly substituted them with visual prototypes during our training sessions. We compare our method with representative FSCIL baselines and recent CLIP-based methods using their reported results.

GMF-CLIP achieves an average accuracy of 82.89%, improving the reported CLIP zero-shot baseline by 12.31 percentage points and the recent QR-Prompt by 3.57 points. The experimental results demonstrate that our method remains applicable in more realistic and complex scenarios.

Table 5: Persistent auxiliary memory on Split ImageNet-R (20-task). The common CLIP ViT-B/16 backbone and the shared replay budget of 1,000 images are excluded. Memory is calculated using FP32 storage. Only methods requiring more auxiliary memory than GMF-CLIP are included.

Method	**Auxiliary** parameters / features	**Memory** (MiB)	**Relative to Ours**
TPPT-V	4.403M	16.80	7.42×
PROOF	11.689M	44.59	19.70×
CLAP4CLIP	14.710M	56.12	24.79×
GMF-CLIP	**0.593M**	**2.26**	**1.00×**

D The Memory and Training Efficiency Analysis

D.1 Memory Overhead Analysis

We analyze the persistent memory required after learning each incremental task. Since all compared methods use the same CLIP backbone, we exclude the shared backbone parameters and focus on the additional memory introduced for continual adaptation.

GMF-CLIP maintains a fixed-size replay buffer of $m = 1{,}000$ images, which is identical to that used by all replay-based baselines in our experiments. Beyond this common replay budget, GMF-CLIP only stores a lightweight gating network, one visual prototype per observed class, and the corresponding CLIP textual features. Importantly, GMF-CLIP does not instantiate a separate prompt, adapter, projection module, or expert for each incremental task. Therefore, its auxiliary model memory does not grow with the number of tasks.

For the gating architecture used in our experiments, consisting of a shared $512 \to 256$ layer, a 1,000-dimensional class-specific output branch, and a scalar sample-specific branch, the gating network contains only $388{,}585$ parameters, corresponding to approximately 1.48 MiB in FP32. The visual and textual prototype banks require $2Cd$ floating-point values, where C is the number of observed classes and $d = 512$ is the CLIP embedding dimension. Even on Split SUN397, caching both banks requires only 1.55 MiB. Consequently, the total persistent overhead beyond the replay buffer and shared CLIP backbone is approximately 3.03 MiB.

This storage pattern compares favorably with methods that continually expand task-specific modules. PROOF introduces new image and text projection modules for each task, CLAP4CLIP maintains task-specific distribution encoders and probabilistic adapters, and TPPT-V incrementally adds visual prompts across multiple Transformer layers. Their parameter storage therefore increases with the number of incremental tasks. In contrast, the model-side overhead of GMF-CLIP remains unchanged when the same set of classes is divided into more tasks. For example, GMF-CLIP has the same final auxiliary model size for the 5-, 10-, and 20-task splits of ImageNet-R, since all three settings contain the same 200 classes.

Compared with replay-based methods using a single non-expanding adapter or classifier, such as CIL-PVLM and MG-CLIP+replay, GMF-CLIP uses the same exemplar budget and introduces only a few megabytes of additional model storage. Exemplar-free approaches such as RAPF naturally avoid the replay-buffer cost; therefore, we do not claim lower total storage than these methods. Nevertheless, GMF-CLIP provides a favorable accuracy–memory trade-off while retaining bounded auxiliary model storage and avoiding task-wise network expansion.

As shown in Table 5, GMF-CLIP introduces only 0.593M auxiliary parameters and stored features, corresponding to 2.26 MiB in FP32. This estimate conservatively includes the entire 1,000-class output capacity of the gating network as well as both the visual and textual feature banks. In comparison, TPPT-V, PROOF, and CLAP4CLIP require 7.42×, 19.70×, and 24.79× more persistent auxiliary memory, respectively.

The advantage is particularly pronounced in long incremental sequences. PROOF and CLAP4CLIP accumulate separate projection or distribution-adapter parameters for each task, whereas GMF-CLIP uses a single shared gating network. Consequently, dividing the same 200 ImageNet-R classes into

Table 6: Computational and wall-clock training efficiency comparison. The common CLIP backbone computation is excluded. Results are estimated on Split ImageNet-R (20-task) with the same batch size and hardware.

Method	**Forward + backward** cost (GMAC)	**Relative cost**	**Relative time** per epoch
CLAP4CLIP	≈ 2.91	$24.0\times$	$\sim 2.5\times$
PROOF	≥ 55.70	$\geq 458.9\times$	$\sim 4.2\times$
GMF-CLIP	$\leq$ **0.121**	**1.0**$\times$	**1.0**$\times$

more incremental tasks does not increase the final auxiliary model size of GMF-CLIP. Moreover, the cached textual features counted above can be deterministically recomputed from the frozen text encoder, making 2.26 MiB a conservative upper bound on its necessary persistent storage.

D.2 Training Efficiency

We evaluate training efficiency using the computational cost of one forward and backward pass through the method-specific auxiliary modules, excluding the shared CLIP backbone.

GMF-CLIP requires at most 0.121 GMACs of auxiliary forward–backward computation per mini-batch. In comparison, CLAP4CLIP and PROOF require approximately $24.0\times$ and at least $458.9\times$ more computation, respectively. This advantage comes from the lightweight deterministic gate of GMF-CLIP, whereas CLAP4CLIP employs a Transformer decoder and 20 Monte Carlo samples, and PROOF performs self-attention over the query, visual prototypes, textual prototypes, and accumulated task prompts.

Besides theoretical computational complexity, we also estimate the wall-clock training overhead. Under the same training configuration, GMF-CLIP introduces negligible additional time compared with the shared CLIP optimization, since the gating network only consists of lightweight linear layers. In contrast, CLAP4CLIP and PROOF require substantially more time per epoch due to stochastic sampling and cross-modal attention operations, respectively.