Memory-Driven Contrastive Embedding Enhancement for Fine-Grained Open-Set Semi-Supervised Learning

Yinan Han[†] **and Qing-Yuan Jiang**[†‡*]
[†]Nanjing University of Science and Technology
[‡]State Key Lab. for Novel Software Technology, Nanjing University, P.R. China
{hanyinan, jiangqy}@njust.edu.cn

Abstract

Existing VLM-based open-set semi-supervised learning (OSSL) methods primarily rely on coarse-grained class-level semantics, thereby hampering the accurate identification of in-distribution (ID) versus out-of-distribution (OOD) samples. To overcome this issue, we propose a novel memory-driven contrastive embedding enhancement approach by integrating fine-grained visual contrastive cues into class-level textual side. Concretely, we design a memory bank to construct and maintain a diverse collection of the most distinctive fine-grained visual embeddings. Guided by the memory bank, visual information is incorporated into the textual side through cross-modal contrastive learning, enabling more discriminative fine-grained separation between ID and OOD samples. Consequently, the learned OSSL model exhibits improved contrastive discriminability across known and unknown categories. Extensive experiments demonstrate that our method achieves state-of-the-art (SOTA) performance on multiple fine-grained datasets. The code is available at https://github.com/njustkmg/NeurIPS26-MDCEE.

1 Introduction

Semi-supervised learning (SSL) [10] is a prominent learning paradigm, as it effectively exploits both labeled and unlabeled data. However, real-world settings are often open-world, rendering the standard SSL assumption of identical training and test distributions invalid. This indicates that unlabeled data in real-world scenarios frequently include a substantial number of previously unseen classes. This problem setting, termed open-set semi-supervised learning (OSSL) [35, 36, 25, 17, 1], has gained increasing research interest in recent years.

To address the challenges of OSSL, extensive efforts have been made in the literature. Early approaches [2, 21, 30] rely on traditional deep learning models, which provide limited semantic priors and are therefore less reliable under open-set noise. Recent advances in foundation models have brought new perspectives to OSSL. Large-model-based approaches [17] enhance the ability to distinguish between in-distribution (ID) and out-of-distribution (OOD) samples by exploiting the rich semantic representations of foundation models such as CLIP [24]. Consequently, they have demonstrated strong effectiveness in OSSL applications.

Despite achieving strong performance, existing approaches [17] usually rely on coarse-grained class-level text representations, limiting their ability to capture fine-grained semantic differences. This issue becomes especially critical in open-world settings, as some OOD categories exhibit only fine-grained differences from ID categories. For example, distinguishing between a *Corgi*, a *Husky*, and a *German Shepherd* cannot be reliably achieved using class-level semantics alone. By incorporating fine-grained attributes such as a *Corgi's short legs* and *golden fur*, the discrimination task becomes substantially

*Qing-Yuan Jiang is the corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

easier, as illustrated in Figure 1. This raises a natural question of how to obtain fine-grained and distinctive representations. Such subtle characteristics are inherently encoded in visual images. Accordingly, a natural solution is to extract fine-grained visual features and leverage them to enhance text-side representation learning, thereby injecting sufficiently discriminative information into textual representations.

Motivated by this insight, in this paper, we introduce a memory-driven contrastive embedding enhancement method that injects fine-grained visual contrastive cues into class-level text representations. First, to preserve fine-grained visual information, we design a class-level memory bank. Each memory vector is updated by incorporating patch-level visual features that are most relevant to the corresponding text cues. As a result, each memory vector aggregates the most discriminative visual cues of its associated class. Then, leveraging the fine-grained visual memory bank, we enhance in-class textual embeddings by incorporating discriminative visual cues, thereby enriching class semantics and improving the model's ability to distinguish subtle differences. Meanwhile, we combine dissimilar visual cues from the memory bank with base textual embeddings to construct a poisoned out-of-class text feature, further strengthening discriminative capability. By constructing these two contrastive text features, we inject rich and fine-grained visual information into textual embeddings, which improves the model's ability to distinguish between in-distribution and out-of-distribution samples during training. Consequently, the proposed approach significantly improves discriminative performance in OSSL settings. Our main contributions are listed as follows:

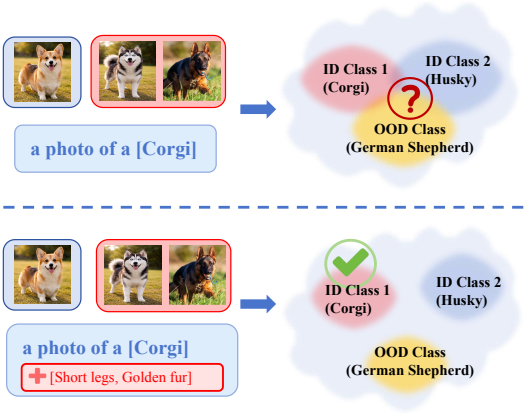


Figure 1: Issue of coarse-grained class-level textual cues. Top: cues with only class names fail to distinguish fine-grained classes and are vulnerable to OOD noise. Bottom: enriched cues with visual attributes (e.g., "short legs, golden fur"), achieving clearer decision boundaries and improved robustness against open-set noise.

- A novel fine-grained visual memory bank is designed, which captures discriminative visual cues that are closely associated with class-level semantics.
- A cross-modal contrastive learning mechanism that supports both in-class and out-of-class learning is proposed. By leveraging a fine-grained visual memory bank, rich visual cues are injected into textual embeddings, thereby improving the model's ability to distinguish between ID and OOD classes.
- Extensive experiments reveal that our proposed method achieves the best performance on various benchmarks compared with SOTA baselines.

2 Related Work

2.1 Open-Set Semi-Supervised Learning

Open-set semi-supervised learning (OSSL) targets a more realistic scenario, where unlabeled datasets inevitably contain OOD samples belonging to unknown categories absent from the labeled set [34, 33, 8, 7]. Early methods for OSSL, such as DS^{3}L [6], proposes a bi-level optimization framework that selectively utilizes unlabeled data by learning instance-wise weights. However, these early approaches suffer from notable limitations, particularly in handling uncertainty. In order to address these shortcomings, recent advancements have introduced increasingly sophisticated techniques. IOMatch [19] employs a multi-binary classifier with a closed-set classifier to produce unified open-set targets, treating all outliers explicitly as one class. The literature [38] proposes Adaptive Negative Evidential Deep Learning (ANEDL), utilizing evidential deep learning to quantify different types of uncertainties in predictions and adaptively optimize negative sampling strategies to differentiate inliers from outliers. CaliMatch [1] proposes an adaptive calibration framework that jointly calibrates the classifier and the OOD detector via adaptive label smoothing and temperature scaling. Recently,

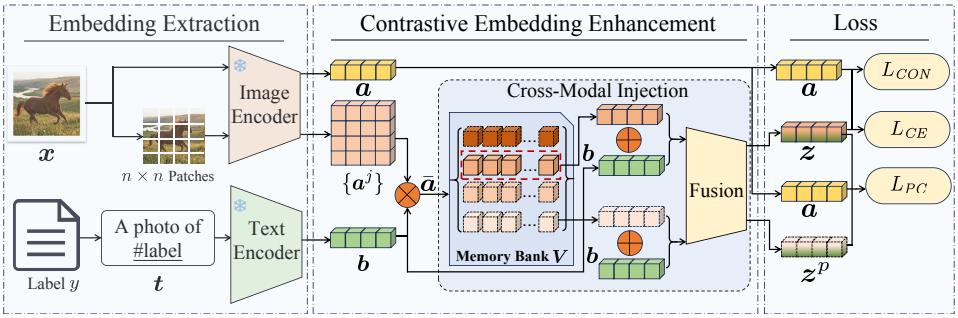


Figure 2: The architecture of our method. We design a **memory bank module** to maintain the most discriminative fine-grained visual cues. Based on this, a **contrastive learning mechanism** is designed to learn embeddings from both in-class and out-of-class perspectives.

CFSG-CLIP [17] introduces VLMs to OSSL. Despite these advancements, existing VLM-based OSSL methods [17] primarily rely on the inherent cross-modal alignment of pre-trained VLMs, where textual embeddings capture only coarse-grained semantic information. This coarse-grained representation is insufficient for distinguishing ID from near-OOD samples in fine-grained scenarios, where subtle visual differences are critical for accurate classification and OOD detection.

2.2 Pre-trained Vision-Language Models

Pre-trained vision-language models (VLMs) learn aligned image-text representations from large-scale image-text pairs, enabling open-vocabulary recognition by similarity matching in a shared embedding space. Representative dual-encoder contrastive pre-training paradigms include CLIP [24] and its scalable variants such as ALIGN [9], while more fine-grained alignment has also been explored by strengthening cross-modal modeling and capturing token-level correspondence [16, 37]. Built upon these backbones, downstream research mainly follows several recurring directions. One line focuses on efficient adaptation through minimal parameter updates or lightweight modules while preserving open-vocabulary capability [11, 40, 31, 5]. Another line extends VLMs from image-level to region-level recognition, learning region-aware representations to detect and ground novel categories beyond the training vocabulary [29, 18]. Despite these advances, existing CLIP-based methods primarily target standard supervised or few-shot scenarios and overlook the unique challenges of OSSL, where unlabeled data contains unknown categories and fine-grained ID-OOD discrimination is critical.

3 Method

In this section, we illustrate our proposed method in detail. The architecture of our method is shown in Figure 2.

3.1 Preliminary

Assume that we have a labeled dataset $\boldsymbol{X}_{\text{labeled}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ denotes the data sample drawn from the distribution $\mathcal{D}_{\text{labeled}}$, and $y_i \in \{1, \cdots, C\}$ indicates the class label with C classes. Furthermore, a large unlabeled dataset $\boldsymbol{X}_{\text{unlabeled}} = \{\boldsymbol{x}_i\}_{i=1}^{M}$ collected from the open-set environment is also available, where sample $\boldsymbol{x}_i$ is drawn from the distribution $\mathcal{D}_{\text{unlabeled}}$, which may contain both ID and OOD data. OSSL aims to correctly classify ID samples while effectively rejecting OOD samples using both the labeled set $\boldsymbol{X}_{\text{labeled}}$ and the unlabeled set $\boldsymbol{X}_{\text{unlabeled}}$.

In recent years, the rapid progress of large-scale pretrained VLMs has substantially advanced OSSL. Most existing OSSL methods [17] leverage the vision and text encoders of pretrained VLMs to obtain modality-specific embeddings. This process can be formulated as $\boldsymbol{a}_i = \phi^v(\boldsymbol{x}_i)$ and $\boldsymbol{b}_i = \phi^t(\boldsymbol{t}_i)$, where $\phi^v(\cdot)$ and $\phi^t(\cdot)$ respectively denote the vision and text encoders, and $\boldsymbol{t}_i$ represents a label-based textual prompt constructed from the class label y_i, e.g., "*A photo of* y_i". Based on the VLM-derived embeddings, the model performs ID-OOD discrimination by comparing samples against class-level textual prototypes.

Under this architecture, the text modality primarily captures coarse-grained visual semantics through cross-modal alignment, which limits its ability to reliably distinguish ID and OOD samples, especially in fine-grained open-set scenarios.

3.2 Visual Memory Bank

To enhance the discriminative capacity of the text modality for ID-OOD separation, we propose a memory bank-driven visual enhancement mechanism.

Given an input image $\boldsymbol{x}_i$, we partition it into n^2 non-overlapping patches, which are subsequently embedded as a sequence of patch-level tokens:

$$\forall j \in \{1, \cdots, n^2\},\ \boldsymbol{a}_i^j = \phi^v(\boldsymbol{x}_i^j) \in \mathbb{R}^d, \tag{1}$$

where $\boldsymbol{x}_i^j$ indicates the j-th patch, and d denotes the feature dimension. Furthermore, the textual embedding $\boldsymbol{b}_i$ is obtained by encoding the label-based textual prompt $\boldsymbol{t}_i$ constructed from y_i.

Based on the patch-level tokens and the textual embedding, we maintain a memory bank $\boldsymbol{V} \in \mathbb{R}^{C \times d}$, with one d-dimensional memory vector for each ID class. We use patch-text similarity as a class-relevance prior. Specifically, we first calculate the similarity score between $\{\boldsymbol{a}_i^j\}_{j=1}^{n^2}$ and $\boldsymbol{b}_i$:

$$\forall j \in \{1, \cdots, n^2\},\ s_i^j = \cos(\boldsymbol{a}_i^j, \boldsymbol{b}_i). \tag{2}$$

Based on the similarity scores, we select the top-K patch-level tokens and aggregate them into an averaged correlated memory slot by an alignment module function $g(\cdot)$:

$$\bar{\boldsymbol{a}}_i = \frac{1}{K} \sum_{k=1}^{K} g(\boldsymbol{a}_i^{r_k}; \theta_g), \tag{3}$$

where $r_k \in \Gamma_K(\{s_i^j\}_{j=1}^{n^2})$, and $\Gamma_K(\{s_i^j\}_{j=1}^{n^2})$ denotes the index set of the top-K patch-level tokens ranked by similarity scores, θ_g denotes the parameters of $g(\cdot)$.

During the training process, the normalized memory slot $\bar{\boldsymbol{a}}_i$ is used to update its corresponding class-wise memory bank $\boldsymbol{v}_{y_i} \in \boldsymbol{V}$ by exponential moving average (EMA) strategy at t-th iteration:

$$\boldsymbol{v}_{y_i}(t+1) = \begin{cases} \bar{\boldsymbol{a}}_i, & \text{if } t = 1, \\ \mu \boldsymbol{v}_{y_i}(t) + (1-\mu) \cdot \dfrac{\bar{\boldsymbol{a}}_i}{\|\bar{\boldsymbol{a}}_i\|}, & \text{otherwise.} \end{cases}$$

By doing so, the memory bank aggregates the most relevant fine-grained visual information, enabling stable and discriminative text-side enhancement.

3.3 Contrastive Embedding Enhancement

We next investigate how fine-grained visual cues can be integrated into the text modality to enhance ID-OOD discrimination. For coarse-grained textual cues, we incorporate fine-grained visual features from the same class to enrich semantic specificity, while simultaneously contrasting them with visually dissimilar features from other classes to promote class separability in the joint embedding space. Accordingly, we design a cross-modal injection strategy that jointly accounts for both in-class relevance and out-of-class discrepancy.

Specifically, we first utilize $\boldsymbol{v}_{y_i}$ to enrich the base textual embedding $\boldsymbol{b}_i$:

$$\boldsymbol{z}_i = \alpha h\left([\boldsymbol{v}_{y_i}; \boldsymbol{b}_i]; \theta_h\right) + \boldsymbol{b}_i, \tag{4}$$

where $[\cdot;\cdot]$ indicates the vector concatenation, $h(\cdot)$ denotes a fusion module function, θ_h denotes the parameters of $h(\cdot)$, and α is a weighting coefficient.

Simultaneously, we inject fine-grained visual cues from different classes into the base textual cue. Specifically, we randomly sample a memory vector from the memory bank corresponding to a class other than y_i:

$$\xi_i \leftarrow \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_C\} \setminus \{\boldsymbol{v}_{y_i}\}. \tag{5}$$

Here we use the notation "$\leftarrow$" to indicate that an element is randomly chosen from a given set. Then the textual embedding $\boldsymbol{b}_i$ is poisoned by:

$$\boldsymbol{z}_i^p = \alpha h\left([\xi_i; \boldsymbol{b}_i]; \theta_h\right) + \boldsymbol{b}_i. \tag{6}$$

By injecting fine-grained visual cues from both similar and dissimilar classes into the textual side, we enable a more robust contrastive learning mechanism. We first calculate the probability given $(\boldsymbol{x}_i, y_i)$ based on the enriched textual embedding $\boldsymbol{z}_i$:

$$p(y_i \mid \boldsymbol{x}_i) = \frac{\exp\left(\langle \boldsymbol{z}_i, \boldsymbol{a}_i \rangle / \tau\right)}{\sum_{j=1}^{C} \exp\left(\langle \boldsymbol{z}_j, \boldsymbol{a}_i \rangle / \tau\right)}, \tag{7}$$

where τ is the temperature parameter. Then the corresponding loss function is defined as:

$$L_{CE}(\boldsymbol{z}_i, \boldsymbol{a}_i) = -\log p(y_i \mid \boldsymbol{x}_i). \tag{8}$$

For unlabeled data, we adopt the standard consistency regularization paradigm with confidence-based filtering to exclude OOD samples from pseudo-labeling. Given an unlabeled sample $\boldsymbol{x}_i \in \boldsymbol{X}_{\text{unlabeled}}$, we apply weak augmentation to obtain $\boldsymbol{x}_i^w$ and strong augmentation to obtain $\boldsymbol{x}_i^s$. We generate a pseudo-label $\hat{y}_i = \operatorname{argmax}_y p(y \mid \boldsymbol{x}_i^w)$ from the weakly-augmented view and compute its prediction confidence. Only samples with confidence exceeding a threshold ρ are treated as ID samples and retained for consistency regularization:

$$\delta_i = \mathbb{1}[p(\hat{y}_i \mid \boldsymbol{x}_i^w) \geq \rho], \tag{9}$$

where $\delta_i \in \{0, 1\}$ is a binary mask indicating whether the sample should contribute to the loss and $\mathbb{1}(\cdot)$ is the indicator function. The consistency regularization loss is then defined as:

$$L_{CON}(\boldsymbol{a}_i^s) = \delta_i \cdot \mathcal{H}(\hat{y}_i, p(y \mid \boldsymbol{x}_i^s)), \tag{10}$$

where $\mathcal{H}(\cdot, \cdot)$ denotes the cross-entropy loss, $\boldsymbol{a}^s$ denotes the embedding extracted from $\boldsymbol{x}^s$.

Then, we calculate the probability based on the poisoned textual embedding $\boldsymbol{z}_i^p$:

$$p^p(y_i \mid \boldsymbol{x}_i) = \frac{\exp\left(\langle \boldsymbol{z}_i, \boldsymbol{a}_i \rangle / \tau\right)}{\exp\left(\langle \boldsymbol{z}_i, \boldsymbol{a}_i \rangle / \tau\right) + \exp\left(\langle \boldsymbol{z}_i^p, \boldsymbol{a}_i \rangle / \tau\right)}. \tag{11}$$

We employ a contrastive loss to encourage image features to align with the correctly enriched embedding $\boldsymbol{z}_i$ while being repelled from the poisoned embedding $\boldsymbol{z}_i^p$:

$$L_{PC}(\boldsymbol{z}_i^p, \boldsymbol{z}_i) = -\log p^p(y_i \mid \boldsymbol{x}_i). \tag{12}$$

By enforcing contrastive alignment, this mechanism encourages the model to focus on distinctive class-specific cues rather than coarse semantics shared across similar classes. Consequently, near-OOD samples that exhibit only partial or superficial similarities to ID classes are naturally assigned lower confidence scores and filtered out by the confidence-based mechanism in Eq. (9), thereby preventing open-set noise from contaminating the training process.

Training Pipeline: The alignment module serves to bridge the modality gap between patch-level visual tokens and textual embeddings, facilitating accurate memory bank updates. To maintain stable semantic correspondence, we optimize the alignment module using the base textual embeddings $\boldsymbol{b}_i$. The corresponding loss function is defined as:

$$\mathcal{L}_{Memory} = \frac{1}{N} \sum_{i=1}^{N} L_{CE}(\boldsymbol{b}_i, \bar{\boldsymbol{a}}_i) + \frac{\lambda}{M} \sum_{i=1}^{M} L_{CON}(\bar{\boldsymbol{a}}_i^s), \tag{13}$$

where λ is a weighting coefficient.

At the same time, based on the memory bank, we optimize the parameters of the fusion module. The corresponding objective function is defined as:

$$\mathcal{L}_{Total} = \frac{1}{N} \sum_{i=1}^{N} \left[L_{CE}(\boldsymbol{z}_i, \boldsymbol{a}_i) + \nu L_{PC}(\boldsymbol{z}_i^p, \boldsymbol{z}_i) \right] + \frac{\lambda}{M} \sum_{i=1}^{M} L_{CON}(\boldsymbol{a}_i^s), \tag{14}$$

where ν is a weighting coefficient.

Algorithm (1) summarizes the entire training process of our proposed method.

Algorithm 1 Algorithm for our approach.

Input: Labeled set $\boldsymbol{X}_{\text{labeled}}$, unlabeled set $\boldsymbol{X}_{\text{unlabeled}}$, encoders $\phi^v(\cdot)$, $\phi^t(\cdot)$;
Output: Memory bank $\boldsymbol{V}$ and model parameters $\theta = \{\theta_g, \theta_h\}$;
INIT: Initialize memory bank $\boldsymbol{V}$, max iterations T.
for $t = 1$ to T **do**
  for each labeled sample $(\boldsymbol{x}_i, y_i)$ **do**
    Extract $\{\boldsymbol{a}_i^j\}_{j=1}^{n^2}$ and $\boldsymbol{b}_i$ based on $\phi^v(\cdot)$, $\phi^t(\cdot)$.
    Calculate similarity of $\{\boldsymbol{a}_i^j\}_{j=1}^{n^2}$ and $\boldsymbol{b}_i$.
    Calculate $\bar{\boldsymbol{a}}_i$ based on Eq. (3).
    Update memory $\boldsymbol{v}_{y_i}$ with $\bar{\boldsymbol{a}}_i$.
    Calculate $\boldsymbol{z}_i$ based on Eq. (4).
    Calculate $\boldsymbol{z}_i^p$ based on Eq. (6).
    Calculate $L_{CE}(\boldsymbol{z}_i, \boldsymbol{a}_i)$ based on Eq. (8).
    Calculate $L_{PC}(\boldsymbol{z}_i^p, \boldsymbol{z}_i)$ based on Eq. (12).
  end for
  for each unlabeled sample $\boldsymbol{x}_i$ **do**
    Generate $\boldsymbol{x}_i^w$, $\boldsymbol{x}_i^s$.
    Calculate $L_{CON}(\boldsymbol{a}_i^s)$ based on Eq. (10).
  end for
  Update θ_g based on Eq. (13).
  Update θ_h based on Eq. (14).
end for
return Memory bank $\boldsymbol{V}$ and model parameters θ;

4 Experiments

4.1 Experimental Setting

Datasets. We evaluate the proposed approach on five widely-used fine-grained benchmark datasets: CUB-200-2011 [28], Stanford Dogs [13], Stanford Cars [14], FGVC-Aircraft [22] and Semi-Aves [27]. We follow the data split strategy from literature [17]. Specifically, we designate the odd classes as in-distribution classes and the even classes as out-of-distribution classes to minimize domain mismatch between ID and OOD categories. For each ID class, we randomly select 5 or 20 samples from the training set as labeled data, using the remaining training samples as unlabeled data. For evaluation, we apply the same ID/OOD class partition to the test set, and evaluate on all test samples from both ID and OOD classes.

Evaluation Metrics. Following previous works [19, 17], we report two key metrics: (1) classification accuracy on ID classes, which measures the model's ability to correctly classify known categories; (2) open-set classification balanced accuracy, which evaluates ID recognition and OOD detection in a unified metric. Specifically, we treat all unseen classes as a single extra category (i.e., the $(C+1)$-th class), and report the macro-averaged recall over all $C+1$ classes. Additionally, we evaluate OOD detection performance using the Area Under the Receiver Operating Characteristic curve (AUROC) and the false positive rate at a true positive rate of 95% (FPR95) [20]. Higher AUROC and lower FPR95 indicate better OOD detection performance.

Baselines. To illustrate the effectiveness of our method on fine-grained OSSL, we compare it with representative vision-language baselines and recent CLIP-driven OSSL approaches. Specifically, we adopt CLIP [24] as the primary baseline and report its zero-shot performance without fine-tuning. In addition to the CLIP baseline, we consider representative prompt-learning baselines such as CoOp [41], PLOT [3], MaPLe [12] and TextRefiner [32]. We also include LoCoOp [23], a prompt-learning method originally proposed for few-shot OOD detection. Additionally, we evaluate parameter-efficient CLIP adaptation methods, including CLIP-Adapter [4] and CLIP-LoRA [39]. For CLIP-based OSSL baselines, we compare against CFSG-CLIP [17], a recent CLIP-driven framework for fine-grained OSSL.

Implementation Details. We adopt a pretrained ViT-B/16 CLIP model as the backbone. We implement the alignment module as an image adapter followed by a two-layer MLP, and the fusion

Table 1: Classification accuracy on ID classes (%) for CLIP-based methods on four fine-grained benchmark datasets with varying labeled set sizes under the OSSL setting.

Method	Stanford Dogs		Stanford Cars		CUB-200-2011		FGVC Aircraft	
	5	20	5	20	5	20	5	20
CLIP	79.25±0.00	79.25±0.00	75.97±0.00	75.97±0.00	66.00±0.00	66.00±0.00	31.37±0.00	31.37±0.00
CLIP-LORA	83.81±0.37	84.31±0.27	82.71±0.56	82.45±1.30	70.95±0.85	73.40±0.75	40.89±1.73	42.37±0.71
CLIP-Adapter	82.91±0.25	86.02±0.27	84.31±0.02	87.13±0.28	80.03±0.29	84.77±1.02	47.77±0.90	55.79±0.73
CoOp	83.01±0.26	85.68±0.37	85.45±0.31	87.64±0.46	80.10±0.29	85.40±0.37	45.39±0.96	55.43±0.30
LoCoOp	83.08±0.25	86.26±0.11	84.10±0.72	87.83±0.66	79.27±0.45	85.63±0.54	45.53±1.36	54.67±1.59
PLOT	84.46±0.07	87.11±0.09	86.28±0.30	88.59±0.45	81.43±0.66	87.20±0.14	49.59±0.37	58.25±0.93
TextRefiner	85.05±0.20	87.38±0.23	87.18±0.29	89.46±0.36	82.35±0.38	88.05±0.28	50.94±0.54	61.80±0.88
MaPLe	85.64±0.15	87.64±0.20	88.16±0.25	90.34±0.25	83.30±0.33	88.77±0.21	52.43±0.47	64.33±1.21
CFSG-CLIP	85.48±0.21	89.42±0.16	90.38±0.09	93.08±0.08	84.73±0.17	91.75±0.24	61.09±0.27	73.56±0.58
Ours	**86.51**±0.18	**90.24**±0.14	**91.19**±0.12	**93.68**±0.43	**88.09**±0.15	**92.59**±0.19	**61.91**±0.32	**73.90**±0.53

Table 2: Open-set classification balanced accuracy (%) for CLIP-based methods on four fine-grained benchmark datasets with varying labeled set sizes under the OSSL setting.

Method	Stanford Dogs		Stanford Cars		CUB-200-2011		FGVC Aircraft	
	5	20	5	20	5	20	5	20
CLIP	77.17±0.00	77.17±0.00	75.70±0.00	75.70±0.00	64.10±0.00	64.10±0.00	31.08±0.00	31.08±0.00
CLIP-LORA	82.34±0.57	82.67±0.36	82.10±0.56	81.03±1.24	70.52±1.34	72.20±0.45	40.10±1.69	41.57±0.69
CLIP-Adapter	81.63±0.14	84.36±0.25	83.65±0.04	86.50±0.25	81.36±0.60	85.20±0.95	46.87±0.88	53.52±1.35
CoOp	81.65±0.20	84.25±0.41	84.63±0.36	86.84±0.42	81.04±0.06	84.92±0.51	44.55±0.94	54.35±0.30
LoCoOp	81.78±0.23	84.68±0.20	83.25±0.76	87.17±0.64	80.45±0.86	85.46±0.32	44.67±1.35	53.60±1.57
PLOT	82.95±0.07	85.54±0.11	85.58±0.32	87.83±0.34	83.32±0.31	87.16±0.17	48.68±0.37	57.13±0.92
TextRefiner	83.52±0.18	85.78±0.24	86.42±0.28	88.65±0.31	84.03±0.41	87.92±0.35	50.20±0.52	60.45±0.84
MaPLe	84.09±0.19	86.02±0.25	87.43±0.27	89.48±0.25	84.72±0.53	88.66±0.60	51.79±0.43	63.12±1.19
CFSG-CLIP	84.02±0.15	87.77±0.19	89.65±0.05	92.34±0.10	86.46±0.25	90.92±0.24	59.92±0.26	72.13±0.57
Ours	**84.89**±0.17	**88.44**±0.22	**90.34**±0.08	**92.62**±0.13	**86.92**±0.31	**91.79**±0.28	**60.70**±0.33	**73.30**±0.49

module as a single-layer network. For the base prompt, we set the learnable textual prompt length to 16, following the setting in CoOp [41]. All models are trained for 51,200 iterations with a batch size of 32 and a learning rate of 0.002. The consistency regularization loss weight λ is set to 2, and the contrastive loss weight ν is set to 1. Following [26], we use a confidence threshold $\rho = 0.95$ to filter reliable pseudo labels for unlabeled samples. The momentum coefficient μ is set to 0.7, the weighting coefficient α is set to 0.3, and the warmup iterations are set to 10,000. The temperature parameter τ is set to 0.5. We set $K = 5$ for patch-level token selection. All results are presented as the mean with standard deviation over three runs using different random seeds.

4.2 Comparison with State-of-the-Art Methods

Table 1 reports the ID classification accuracy of CLIP-based methods on four fine-grained benchmarks with varying labeled set sizes. Our approach consistently achieves the best results across all datasets and label regimes. Notably, on CUB-200-2011 with only 5 labeled samples per class, we achieve 88.09% accuracy, surpassing the strongest prior approach CFSG-CLIP (84.73%) by 3.36%. We observe similarly consistent gains on other datasets, e.g., our approach improves over CFSG-CLIP by 1.03% on Stanford Dogs (5 samples: 86.51% vs. 85.48%) and achieves the best performance on FGVC Aircraft under both 5 samples (61.91%) and 20 samples (73.90%) settings. These results validate the effectiveness of our memory-driven contrastive embedding enhancement approach. By aggregating fine-grained visual features into a class-wise memory bank and refining textual embeddings with these discriminative cues, our approach achieves superior performance compared to methods relying solely on coarse-grained class-level semantics.

Table 2 summarizes the open-set classification balanced accuracy on four fine-grained benchmarks under the same OSSL setting. Our approach achieves the best balanced accuracy across all datasets and label regimes, demonstrating superior open-set recognition capability under limited supervision. On FGVC Aircraft with 20 labeled samples per class, our approach reaches 73.30% balanced accuracy, outperforming CFSG-CLIP (72.13%) by 1.17%. The gains persist in the 5 samples setting, where our approach achieves 60.70% balanced accuracy compared to 59.92% for CFSG-CLIP. These improvements demonstrate the effectiveness of the approach in strengthening ID-OOD discrimination.

Table 3: Classification accuracy (%) on Semi-Aves under two unlabeled-data settings: ID samples only (U_{in}), and a mixture of ID and OOD samples ($U_{in} + U_{out}$).

Method	U_{in}	$U_{in} + U_{out}$
CLIP	$10.05_{\pm 0.00}$	$10.05_{\pm 0.00}$
CLIP-Adapter	$50.16_{\pm 1.55}$	$50.10_{\pm 0.44}$
CoOp	$47.67_{\pm 0.97}$	$47.93_{\pm 0.76}$
LoCoOp	$48.04_{\pm 0.96}$	$47.30_{\pm 1.30}$
PLOT	$53.68_{\pm 0.35}$	$54.72_{\pm 0.96}$
MaPLe	$60.39_{\pm 0.10}$	$59.68_{\pm 0.64}$
CFSG-CLIP	$65.63_{\pm 0.27}$	$63.69_{\pm 0.74}$
Ours	$\mathbf{67.21}_{\pm \mathbf{0.42}}$	$\mathbf{65.27}_{\pm \mathbf{0.68}}$

Table 4: AUROC and FPR95 (%) on four fine-grained datasets.

Dataset	Labels	AUROC ↑		FPR95 ↓	
		CFSG	Ours	CFSG	Ours
Dogs	5	67.35	**77.48**	88.33	**79.68**
	20	69.29	**77.65**	86.87	**78.83**
Cars	5	63.01	**79.84**	89.67	**76.51**
	20	64.29	**82.35**	88.13	**71.31**
CUB	5	65.78	**76.45**	87.96	**81.47**
	20	71.84	**78.95**	82.17	**77.28**
Aircraft	5	50.16	**53.39**	95.26	**94.57**
	20	54.29	**62.49**	94.98	**90.77**

By contrasting enriched textual embeddings against poisoned ones, our model learns distinctive class-specific features that enable reliable separation between ID and near-OOD samples.

We further evaluate our method on Semi-Aves under two unlabeled-data settings: ID samples only (U_{in}) and a mixture of ID and OOD samples ($U_{in} + U_{out}$). As shown in Table 3, our method achieves the best classification accuracy in both settings, reaching 67.21% and 65.27%, respectively.

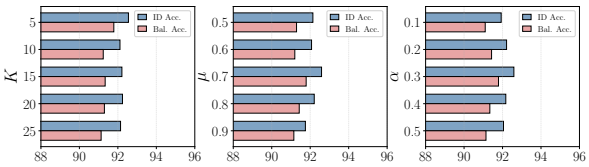
To directly assess OOD detection performance, we report two standard OOD detection metrics, AUROC and FPR95, in Table 4. We compare our method against CFSG-CLIP, the strongest baseline for fine-grained OSSL. Dogs, Cars, CUB, and Aircraft denote Stanford Dogs, Stanford Cars, CUB-200-2011, and FGVC-Aircraft, respectively. Our method consistently achieves higher AUROC and lower FPR95 across all four datasets under both the 5-label and 20-label settings.

4.3 Sensitivity to Hyper-Parameters

We analyze the sensitivity to key hyper-parameters on the CUB-200-2011 dataset with 20 labeled samples per class. We report the classification accuracy on ID classes (ID Acc.) and the open-set classification balanced accuracy (Bal. Acc.) in Figure 3.

Sensitivity to K. As shown in Figure 3 (a), using $K = 5$ achieves the best performance. The performance remains relatively stable across different K values (K from 5 to 25), demonstrating the robustness of our patch-level token selection mechanism.

Sensitivity to μ. Figure 3 (b) shows that our method is robust to the momentum coefficient μ across a wide range of values. Performance remains stable at $\mu \in [0.5, 0.8]$ with ID accuracy consistently above 92%, demonstrating that the visual memory bank updates are not sensitive to the momentum coefficient μ.



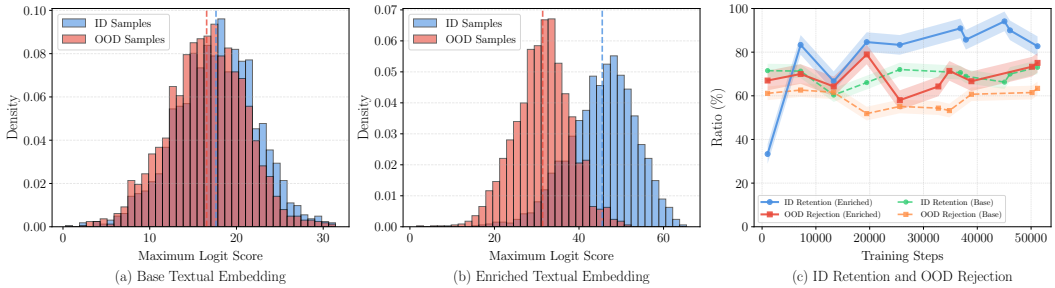
(a) Sensitivity to K (b) Sensitivity to μ (c) Sensitivity to α

Figure 3: Sensitivity analysis on key hyper-parameters. We report results under different settings of K, μ, and α.

Sensitivity to α. Figure 3 (c) demonstrates robustness to the weighting coefficient α. Our approach maintains strong performance across $\alpha \in [0.2, 0.5]$ with ID accuracy above 92%, indicating that the cross-modal injection strategy is not sensitive to the weighting coefficient α.

4.4 Ablation Study

To validate the effectiveness of each component in our approach, we conduct comprehensive ablation studies on the CUB-200-2011 dataset with 20 labeled samples per class. We report the classification accuracy on ID classes (ID Acc.) and the open-set classification balanced accuracy (Bal. Acc.) in Table 5. AM and MU refer to the alignment module and the memory update strategy, respectively.



(a) Base Textual Embedding (b) Enriched Textual Embedding (c) ID Retention and OOD Rejection

Figure 4: ID-OOD separation on CUB-200-2011. (a) Distribution of maximum prediction logits using base textual embeddings. (b) Distribution of maximum prediction logits using enriched textual embeddings. (c) Training dynamics of ID retention and OOD rejection ratios.

Effect of Loss Functions. We progressively incorporate our proposed loss functions to validate their effectiveness. As shown in Rows 1-3 of Table 5, each component consistently improves the final performance. Incorporating the consistency regularization loss L_{CON} increases ID accuracy from 87.80% to 91.27% and balanced accuracy from 86.85% to 90.30%. Further incorporating the contrastive loss L_{PC} yields additional gains, reaching 92.59% ID accuracy and 91.79% balanced accuracy, suggesting that explicitly contrasting semantically consistent versus inconsistent pairs strengthens ID separability and OOD detection.

Table 5: Ablation study on CUB-200-2011 with 20 labeled samples per class.

L_{CE}	L_{CON}	L_{PC}	AM	MU	ID Acc.	Bal. Acc.
✓			✓	✓	87.80	86.85
✓	✓		✓	✓	91.27	90.30
✓	✓	✓	✓	✓	**92.59**	**91.79**
✓				✓	86.64	84.97
✓	✓			✓	88.82	87.42
✓	✓	✓		✓	90.93	90.15
✓			✓		87.35	86.45
✓	✓		✓		89.53	88.55
✓	✓	✓	✓		91.69	90.62

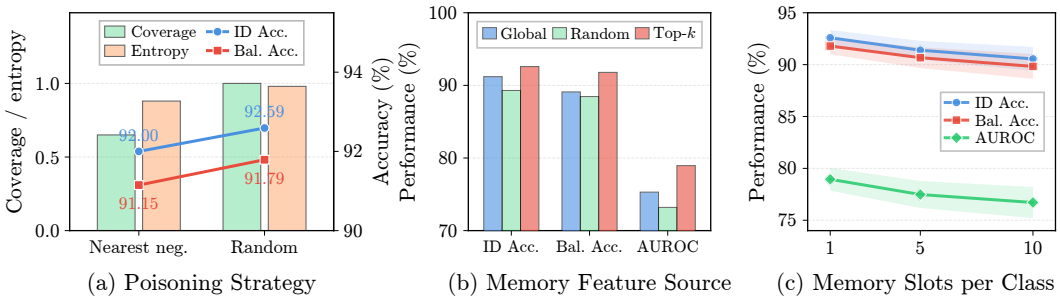
Effect of the Alignment Module. Rows 4-6 of Table 5 examine the role of the alignment module $g(\cdot)$ in bridging the modality gap between patch-level visual tokens and textual embeddings. Without the alignment module, the model achieves 86.64% ID accuracy and 84.97% balanced accuracy using only L_{CE}. Adding L_{CON} improves the performance to 88.82% ID accuracy and 87.42% balanced accuracy, while further incorporating L_{PC} yields 90.93% and 90.15%, respectively. Compared with the full model (Row 3), the remaining gap indicates that the alignment module is critical for effective image-text alignment, bringing a gain of approximately 1.66% in ID accuracy and 1.64% in balanced accuracy. Comparing Rows 1-3 with Rows 4-6 further shows that the alignment module consistently improves both metrics under all three combinations of loss terms.

Effect of Memory Update Strategy. Rows 7-9 of Table 5 evaluate the contribution of memory update strategy. When the memory bank is updated only with labeled samples, the model achieves 91.69% ID accuracy and 90.62% balanced accuracy. Compared with the full model that additionally incorporates high-confidence unlabeled samples for memory updates (Row 3), these results indicate that enriching the memory bank with diverse unlabeled data enables more comprehensive feature aggregation, leading to a gain of approximately 1.2% in balanced accuracy.

4.5 Further Analysis

To better understand how the memory bank improves the distinction between ID and OOD samples, we visualize two aspects in Figure 4: (1) the distribution of maximum prediction logits with base textual embeddings $\{\boldsymbol{b}_i\}$ versus enriched textual embeddings $\{\boldsymbol{z}_i\}$ on the test set, and (2) the training dynamics of ID retention and OOD rejection ratios on the unlabeled dataset. Stronger separation between ID and OOD distributions indicates more effective OOD detection.

As shown in Figure 4 (a), using base textual embeddings results in substantial overlap between ID and OOD distributions, indicating that generic text templates lack sufficient discriminative information to separate the two. In contrast, Figure 4 (b) demonstrates that using enriched textual embeddings



(a) Poisoning Strategy (b) Memory Feature Source (c) Memory Slots per Class

Figure 5: Analysis of poisoning strategies and visual memory bank design on CUB-200-2011 under the 20-label setting. (a) Comparison of random and nearest-negative sampling for constructing poisoned textual embeddings. (b) Comparison of global image features, random patches, and top-k patches for memory construction. (c) Effect of the number of memory slots per class.

achieves much clearer separation, with distinctly separated distributions for ID and OOD samples. Furthermore, Figure 4 (c) reveals that during training, using enriched textual embeddings maintains higher ID retention and OOD rejection ratios than using base embeddings, demonstrating that the memory bank effectively injects discriminative visual information into textual representations.

We further compare different poisoning strategies and visual memory bank designs in Figure 5. Figure 5(a) compares random and nearest-negative sampling for constructing poisoned textual embeddings. Random sampling achieves higher negative-class coverage and sampling entropy, exposing the model to more diverse visual cues and providing richer contrastive signals during training. The higher ID and balanced accuracy further support the effectiveness of diverse negative sampling. Figure 5(b) compares global image features, random patches, and top-K patches for memory construction. Top-K patches perform best across all three metrics, supporting the use of patch-text similarity as a class-relevance prior for memory construction. Figure 5(c) evaluates the effect of the number of memory slots per class. A single slot consistently outperforms five or ten slots, supporting our single-slot memory design.

5 Conclusion

We propose a memory-driven contrastive embedding enhancement approach for fine-grained OSSL, targeting the coarse class-wise semantics that hinder fine-grained ID classification and OOD detection. We design a class-level memory bank that maintains discriminative visual cues for each category. Guided by the memory bank, we inject visual information into textual embeddings through cross-modal contrastive learning, thereby enriching class semantics with fine-grained attributes and improving both ID classification and OOD rejection under open-set noise. Extensive experiments on multiple fine-grained benchmarks demonstrate that our method consistently outperforms strong baselines and achieves state-of-the-art performance.

Limitation. Our visual memory bank is particularly effective when discriminative cues are concentrated in local regions, such as beak shapes or grille designs, which can be captured by top-K patch selection. However, its benefits may provide smaller gains when inter-class differences primarily involve global shapes, such as body proportions or object silhouettes. Incorporating complementary global structural cues into the memory mechanism remains a direction for future work.

Acknowledgments and Disclosure of Funding

This work was supported in part by the NSFC (62506168, 62676195), in part by the Natural Science Foundation of Jiangsu Province of China under Grant (BK20251431, BK20240081), and in part by the Fundamental Research Funds for the Central Universities (No.30925010205).

References

[1] J. Bae, S. B. Kim, and H. Do. Calimatch: Adaptive calibration for improving safe semi-supervised learning. In *ICCV*, pages 2867–2876. IEEE, 2025.

[2] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *AAAI*, pages 6912–6920. AAAI Press, 2021.

[3] G. Chen, W. Yao, X. Song, X. Li, Y. Rao, and K. Zhang. PLOT: prompt learning with optimal transport for vision-language models. In *ICLR*, page online. OpenReview.net, 2023.

[4] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao. Clip-adapter: Better vision-language models with feature adapters. *IJCV*, 132(2):581–595, 2024.

[5] W. Ge, P. Huang, R. Yan, H. Qu, G. Xie, and X. Shu. Reliable and diverse hierarchical adapter for zero-shot video classification. In *IJCAI*, pages 1026–1034. ijcai.org, 2025.

[6] L. Guo, Z. Zhang, Y. Jiang, Y. Li, and Z. Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *ICML*, pages 3897–3906. PMLR, 2020.

[7] Y. Han and Q.-Y. Jiang. PAF: perturbation-aware filtering for open-set semi-supervised learning. In *CVPR*, pages 24803–24812. CVF, 2026.

[8] J. Hang and M. Zhang. Binary decomposition: A problem transformation perspective for open-set semi-supervised learning. In *ICML*, pages 17505–17518. PMLR, 2024.

[9] C. Jia, Y. Yang, Y. Xia, Y. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, volume 139, pages 4904–4916. PMLR, 2021.

[10] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209. Morgan Kaufmann, 1999.

[11] A. Karmanov, D. Guan, S. Lu, A. El-Saddik, and E. P. Xing. Efficient test-time adaptation of vision-language models. In *CVPR*, pages 14162–14171. IEEE, 2024.

[12] M. U. Khattak, H. A. Rasheed, M. Maaz, S. H. Khan, and F. S. Khan. Maple: Multi-modal prompt learning. In *CVPR*, pages 19113–19122. IEEE, 2023.

[13] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshops*, page online, 2011.

[14] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, pages 554–561. IEEE Computer Society, 2013.

[15] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

[16] J. Li, R. R. Selvaraju, A. Gotmare, S. R. Joty, C. Xiong, and S. C. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, pages 9694–9705. Curran Associates, Inc., 2021.

[17] X. Li, Y. Huang, and Q. Guan. Clip-driven coarse-to-fine semantic guidance for fine-grained open-set semi-supervised learning. In *CVPR*, pages 30312–30321. CVF / IEEE, 2025.

[18] Y. Li, J. Niu, and T. Ren. Benefit from seen: Enhancing open-vocabulary object detection by bridging visual and textual co-occurrence knowledge. In *ICCV*, pages 22110–22119, 2025.

[19] Z. Li, L. Qi, Y. Shi, and Y. Gao. Iomatch: Simplifying open-set semi-supervised learning with joint inliers and outliers utilization. In *ICCV*, pages 15824–15833. IEEE, 2023.

[20] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, page online. OpenReview.net, 2018.

[21] Q. Ma, J. Gao, B. Zhan, Y. Guo, J. Zhou, and Y. Wang. Rethinking safe semi-supervised learning: Transferring the open-set problem to A close-set one. In *ICCV*, pages 16324–16333. IEEE, 2023.

[22] S. Maji, E. Rahtu, J. Kannala, M. B. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013.

[23] A. Miyai, Q. Yu, G. Irie, and K. Aizawa. Locoop: Few-shot out-of-distribution detection via prompt learning. In *NeurIPS*, pages 76298–76310. Curran Associates, Inc., 2023.

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pages 8748–8763. PMLR, 2021.

[25] Y. Ren, C. Feng, X. Xie, and S. K. Zhou. Partial optimal transport based out-of-distribution detection for open-set semi-supervised learning. In *IJCAI*, pages 4851–4859. ijcai.org, 2024.

[26] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. Raffel, E. D. Cubuk, A. Kurakin, and C. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, pages 596–608. Curran Associates, Inc., 2020.

[27] J. Su, Z. Cheng, and S. Maji. A realistic evaluation of semi-supervised learning for fine-grained classification. In *CVPR*, pages 12966–12975. CVF / IEEE, 2021.

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[29] Y. Wang, X. Wu, and S. Yang. METOR: A unified framework for mutual enhancement of objects and relationships in open-vocabulary video visual relationship detection. In *IJCAI*, pages 2000–2008. ijcai.org, 2025.

[30] Z. Wang, L. Xiang, L. Huang, J. Mao, L. Xiao, and T. Yamasaki. SCOMatch: Alleviating Overtrusting in Open-Set Semi-supervised Learning. In *ECCV*, pages 217–233. Springer, 2024.

[31] S. Wu, J. Zhang, P. Zeng, L. Gao, J. Song, and H. T. Shen. Skip tuning: Pre-trained vision-language models are effective and efficient adapters themselves. In *CVPR*, pages 14723–14732. CVF / IEEE, 2025.

[32] J. Xie, Y. Zhang, J. Peng, Z. Huang, and L. Cao. Textrefiner: Internal visual feature as efficient refiner for vision-language models prompt tuning. In *AAAI*, pages 8718–8726. AAAI Press, 2025.

[33] H. Xu and Y. Yang. ITP: instance-aware test pruning for out-of-distribution detection. In *AAAI*, pages 21743–21751. AAAI Press, 2025.

[34] H. Xu and Y. Yang. Shaping parameter contribution patterns for out-of-distribution detection. In *AAAI*, pages 27242–27250. AAAI Press, 2026.

[35] Y. Yang, N. J. Jiang, Y. Xu, and D. Zhan. Robust semi-supervised learning by wisely leveraging open-set data. *TPAMI*, 46(12):8334–8347, 2024.

[36] Y. Yang, H. Wei, Z. Sun, G. Li, Y. Zhou, H. Xiong, and J. Yang. S2OSC: A holistic semi-supervised approach for open set classification. *TKDD*, 16(2):34:1–34:27, 2022.

[37] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, and C. Xu. FILIP: fine-grained interactive language-image pre-training. In *ICLR*, page online. OpenReview.net, 2022.

[38] Y. Yu, D. Deng, F. Liu, Q. Dou, Y. Jin, G. Chen, and P. Heng. ANEDL: adaptive negative evidential deep learning for open-set semi-supervised learning. In *AAAI*, pages 16587–16595. AAAI Press, 2024.

[39] M. Zanella and I. B. Ayed. Low-rank few-shot adaptation of vision-language models. In *CVPR Workshops*, pages 1593–1603. IEEE, 2024.

[40] Y. Zhang, Y. Deng, M. Guo, and S. Hu. Adaptive parameter selection for tuning vision-language models. In *CVPR*, pages 4280–4290. CVF/ IEEE, 2025.

[41] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022.

A An Alternative Use of the Visual Memory Bank: Pseudo-OOD Synthesis

In this section, we explore an alternative way to leverage the visual memory bank for enhancing ID-OOD discrimination. Beyond the contrastive embedding enhancement presented in the main paper, the memory bank can also be utilized to synthesize pseudo-OOD samples.

A.1 Pseudo-OOD Construction

Given a labeled sample $(\boldsymbol{x}_i, y_i)$ with its corresponding visual embedding $\boldsymbol{a}_i = \phi^v(\boldsymbol{x}_i)$ and the class-wise memory bank $\boldsymbol{V} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_C\}$, we construct a pseudo-OOD sample by mixing $\boldsymbol{a}_i$ with a memory vector from a different class. Specifically, we first randomly select a class $c \neq y_i$, and then synthesize a pseudo-OOD feature as:

$$\tilde{\boldsymbol{a}}_i = \beta \cdot \boldsymbol{a}_i + (1 - \beta) \cdot \boldsymbol{v}_c, \tag{15}$$

where $\beta \in [0, 1]$ is a mixing coefficient that controls the degree of semantic conflict. $\tilde{\boldsymbol{a}}_i$ intentionally contains conflicting visual semantics from two distinct classes. This semantic conflict simulates a key property of near-OOD samples: they often exhibit partial similarities to multiple ID classes while not belonging to any of them, making them difficult to filter out as they may receive high confidence from the model.

A.2 Uniform Distribution Loss

To encourage the model to assign low and uniform confidence scores to pseudo-OOD samples, we employ a uniform distribution loss. Specifically, given the synthesized feature $\tilde{\boldsymbol{a}}_i$ and the enriched textual embeddings $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_C\}$, we compute the predicted probability distribution:

$$p(j \mid \tilde{\boldsymbol{a}}_i) = \frac{\exp(\langle \tilde{\boldsymbol{a}}_i, \boldsymbol{z}_j \rangle / \tau)}{\sum_{k=1}^{C} \exp(\langle \tilde{\boldsymbol{a}}_i, \boldsymbol{z}_k \rangle / \tau)}, \quad j \in \{1, \ldots, C\}, \tag{16}$$

where τ is the temperature parameter.

The uniform distribution loss is then defined as the cross-entropy between a uniform target distribution and the predicted distribution:

$$L_{UNI}(\tilde{\boldsymbol{a}}_i) = -\sum_{j=1}^{C} \frac{1}{C} \log p(j \mid \tilde{\boldsymbol{a}}_i). \tag{17}$$

By minimizing L_{UNI}, the model learns to avoid making confident predictions on semantically conflicting samples, thereby naturally filtering out near-OOD samples during training.

A.3 Training Integration

The uniform distribution loss is incorporated into the objective for updating the fusion module with a weighting coefficient λ_{ood}:

$$\mathcal{L}_{Total} = \frac{1}{N} \sum_{i=1}^{N} \Big[L_{CE}(\boldsymbol{z}_i, \boldsymbol{a}_i) + \lambda_{\text{ood}} L_{UNI}(\tilde{\boldsymbol{a}}_i) \Big] + \frac{\lambda}{M} \sum_{i=1}^{M} L_{CON}(\boldsymbol{a}_i^s). \tag{18}$$

where L_{CE} and L_{CON} are defined in the main paper, and the alignment module is optimized using the same objective as the main paper.

We validate this pseudo-OOD synthesis module in Appendix C.2. Note that this module is used only for additional analysis and is not used in the main experiments.

B Notation and Datasets

In this section, we summarize the key notation used throughout the paper and provide additional details on the datasets used in our experiments.

Table I: Notation Definition.

Symbol	Description
$\boldsymbol{X}_{\text{labeled}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$	Labeled training set with N samples
$\boldsymbol{X}_{\text{unlabeled}} = \{\boldsymbol{x}_i\}_{i=1}^{M}$	Unlabeled set with M samples
$\mathcal{D}_{\text{labeled}}$	Data distributions for labeled sets
$\mathcal{D}_{\text{unlabeled}}$	Data distributions for unlabeled sets
$\boldsymbol{x}_i$	Input image sample
C	the number of ID classes
$y_i \in \{1, \ldots, C\}$	ID class label
$\phi^v(\cdot)$	Image encoder
$\phi^t(\cdot)$	Text encoder
$\boldsymbol{t}_i$	Label-based textual prompt for class y_i
$\boldsymbol{a}_i$	Global image embedding of x_i
$\boldsymbol{b}_i$	Base textual embedding of prompt t_i
n^2	Number of non-overlapping image patches
$\boldsymbol{x}_i^j$	The j-th patch of image x_i
$\boldsymbol{a}_i^j$	Patch-level token
K	Number of selected top patches
$g(\cdot;\theta_g)$	Alignment module with parameters θ_g
$h(\cdot;\theta_h)$	Fusion module with parameters θ_h
$\boldsymbol{V}$	Visual memory bank
$\boldsymbol{v}_{y_i}$	Memory vector of class y_i
μ	Momentum coefficient
$[\cdot;\cdot]$	Vector concatenation operator
α	Weighting coefficient
$\boldsymbol{z}_i$	Enriched textual embedding
ξ_i	Randomly sampled memory vector
$\boldsymbol{z}_i^p$	Poisoned textual embedding
τ	Temperature parameter
ρ	Confidence threshold for filtering unlabeled samples
$\mathbb{1}(\cdot)$	Indicator function
$H(\cdot,\cdot)$	Cross-entropy operator
L_{CE}	Cross-entropy loss
L_{CON}	Consistency regularization loss
L_{PC}	contrastive loss
λ	Weighting coefficient for L_{CON}
ν	Weighting coefficient for L_{PC}

B.1 Notation

Table I summarizes the key notation used throughout the paper.

B.2 Datasets

Stanford Dogs. Stanford Dogs [13] is a fine-grained dog categories recognition benchmark containing 20,580 images from 120 dog breeds. Following the standard split, it provides 12,000 training images and 8,580 test images, and includes breed labels and bounding-box annotations for the training set. We use the official train/test split for all experiments.

Stanford Cars. Stanford Cars [14] consists of 16,185 images from 196 car categories. The dataset is officially split into 8,144 training images and 8,041 testing images, with an approximately balanced per-class partition. We use the official train/test split for all experiments.

CUB-200-2011. CUB-200-2011 [28] is a widely used fine-grained bird classification dataset with 200 bird categories and 11,788 images in total. The official split contains 5,994 training images and 5,794 testing images. In addition to category labels, it provides rich instance-level annotations, including one bounding box per image, 15 part locations, and 312 binary attributes. We use the official train/test split for all experiments.

FGVC-Aircraft. FGVC-Aircraft [22] is a fine-grained aircraft recognition benchmark comprising 10,000 images across 100 aircraft categories, with each image annotated by a tight bounding box and a hierarchical aircraft model label. The standard protocol partitions images into train/validation/test subsets with roughly equal size per variant (typically $33/33/34$ images per category). We use the official train/test split for all experiments.

C Additional Experimental Results

C.1 Visualization of Selected Patches

To provide insights into what fine-grained visual cues are captured by our visual memory bank, we visualize the top-K selected patches on random samples from four fine-grained datasets (in practice, we set $K = 5$): CUB-200-2011 [28], Stanford Cars [14], Stanford Dogs [13] and FGVC-Aircraft [22]. Figure I shows that the selected patches tend to focus on discriminative local regions that are characteristic of each category.

For bird species (CUB-200-2011) [28], the model selects patches corresponding to distinctive features such as beak shape, eye rings, and plumage patterns. For car models (Stanford Cars) [14], the selected patches concentrate on headlights, grilles, and body contours. For dog breeds (Stanford Dogs) [13], the model focuses on facial features, ear shapes, and fur textures. For aircraft variants (FGVC-Aircraft) [22], the selected patches primarily attend to the fuselage body, cockpit and window regions. These visualizations indicate that our visual memory bank effectively aggregates the most distinctive visual cues for each category.

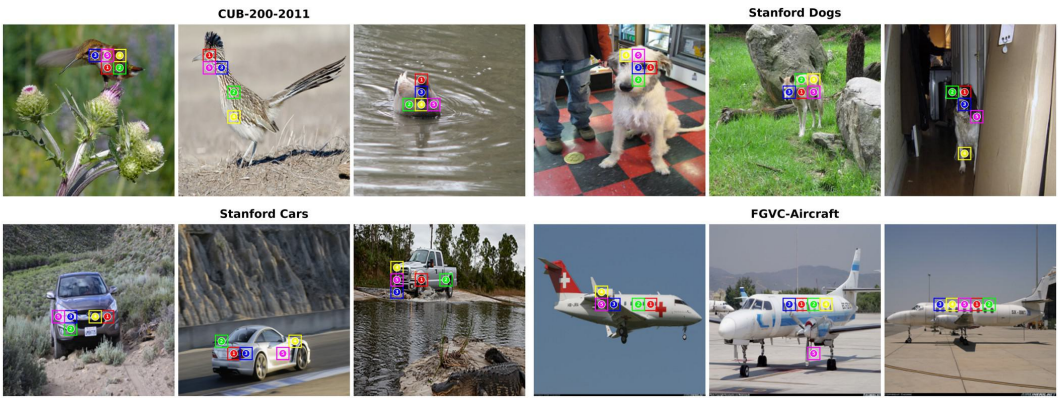


Figure I: Visualization of top-5 selected patches on representative samples. The colored boxes indicate the patches with highest cosine similarity to the textual embedding, numbered by their ranking.

C.2 Impact of Pseudo-OOD Synthesis

To validate the effectiveness of the memory-driven pseudo-OOD synthesis module (Appendix A), we disable the contrastive embedding enhancement component used in the main paper and rely solely on the memory bank for pseudo-OOD synthesis. We then compare models trained with and without this mechanism. We set the mixing coefficient $\beta = 0.5$ and the weighting coefficient $\lambda_{\text{ood}} = 0.1$.

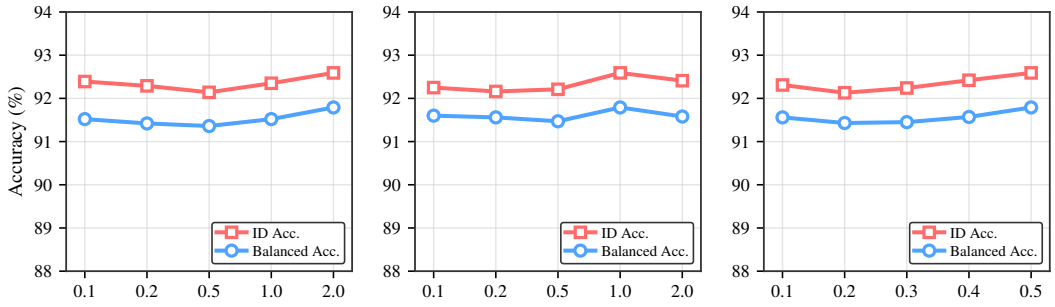
We report the ID classification accuracy (ID Acc.) and the open-set classification balanced accuracy (Bal. Acc.) in Table II. Incorporating pseudo-OOD synthesis brings consistent improvements across all four datasets. Specifically, it increases the ID classification accuracy by roughly 0.24%-1.17% and also improves balanced accuracy by about 0.50%-1.88%. This suggests that semantically conflicting pseudo-OOD training improves both OOD rejection and ID discrimination, yielding more reliable open-set decision boundaries.

C.3 Sensitivity to Hyper-parameters

We report additional sensitivity analyses for three hyper-parameters: the weighting coefficient λ for consistency regularization loss, ν for contrastive loss, and temperature parameter τ. All experiments

Table II: Effect of memory-driven pseudo-OOD synthesis on four datasets (20 labeled samples).

Dataset	Method	ID Acc. (%)	Bal. Acc. (%)
Stanford Dogs	w/o Pseudo-OOD	89.23	87.55
	w/ Pseudo-OOD	**89.92**	**88.05**
Stanford Cars	w/o Pseudo-OOD	92.33	91.46
	w/ Pseudo-OOD	**93.25**	**92.20**
CUB-200-2011	w/o Pseudo-OOD	91.27	90.30
	w/ Pseudo-OOD	**91.51**	**91.28**
FGVC-Aircraft	w/o Pseudo-OOD	72.28	70.92
	w/ Pseudo-OOD	**73.45**	**72.80**



(a) Weighting Coefficient λ (b) Weighting Coefficient ν (c) Temperature Parameter τ

Figure II: Sensitivity analysis of hyper-parameters on CUB-200-2011 (20 labeled samples).

are conducted on the CUB-200-2011 dataset under the 20-labeled setting. As shown in Figure II, our method remains stable over a broad range of hyper-parameters. The results indicate that our approach is not sensitive to hyper-parameters.

C.4 Computational Complexity

All experiments in the main paper are conducted on a single NVIDIA RTX 4090 GPU. We further compare the computational overhead of our method with the strongest baseline, CFSG-CLIP [17], on CUB-200-2011 [28] under the 20-label setting. As shown in Table III, our method improves the ID classification accuracy from 91.75% to 92.59% while maintaining competitive computational cost, with comparable training time and lower memory usage than CFSG-CLIP.

Table III: Computational overhead comparison on CUB-200-2011 under the 20-label setting.

Method	ID Acc. (%)	Extra Params (M)	Train Time (h)	Train Mem (GB)	Infer Mem (GB)
CLIP	66.00	0.0	–	–	3.4
CFSG-CLIP	91.75	0.9	11.0	12.3	4.2
Ours	**92.59**	1.2	12.5	7.2	3.8

C.5 Textual Discriminability Analysis

To understand how the memory bank enhances textual discriminability, we visualize inter-class similarity matrices as heatmaps in Figure III. For clarity, we randomly sample 6 ID classes from CUB-200-2011 [28].

We compare three types of representations: (1) visual memory bank feature vectors $\boldsymbol{v}_{y_i}$, (2) base textual embeddings $\boldsymbol{b}_c$, and (3) enriched textual embeddings $\boldsymbol{z}_i$ that incorporate visual cues from the

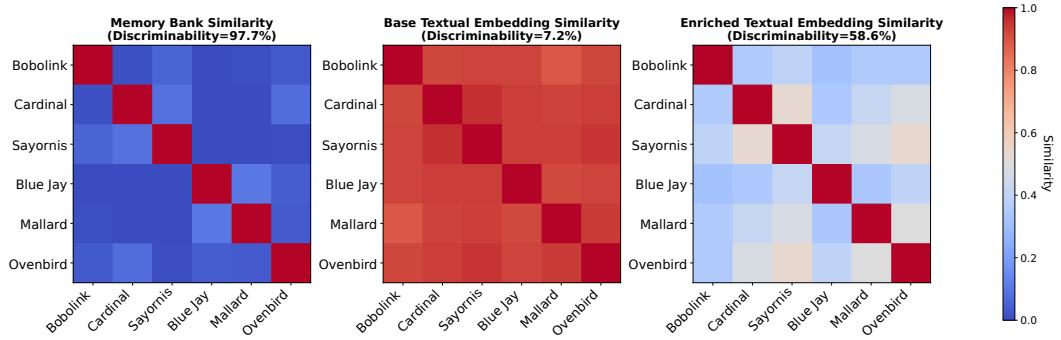


Figure III: Textual discriminability analysis on CUB-200-2011. Left: similarity matrix for visual memory bank feature vectors. Middle: similarity matrix for base textual embeddings. Right: similarity matrix for enriched textual embeddings.

memory bank. We quantify discriminability using the average off-diagonal cosine dissimilarity, expressed as a percentage. Higher values indicate better inter-class separation. Base textual embeddings exhibit high inter-class similarity, with a dissimilarity of only 7.2%. In contrast, visual memory bank features exhibit substantially lower inter-class similarity, with a dissimilarity of 97.7%. Incorporating these visual cues increases the dissimilarity of textual embeddings to 58.6%, demonstrating improved inter-class separation after memory enrichment.

We further extend the quantitative analysis to the other three datasets. As shown in Table IV, enriched textual embeddings consistently exhibit higher inter-class dissimilarity than base textual embeddings.

Table IV: Average off-diagonal cosine dissimilarity (%) of base and enriched textual embeddings across three datasets. Higher values indicate better inter-class separation.

Dataset	Base textual embeddings	Enriched textual embeddings
Stanford Dogs	15.20	**47.00**
Stanford Cars	8.40	**54.76**
FGVC-Aircraft	4.81	**53.58**

C.6 Near- and Far-OOD Detection on CIFAR-100

To directly assess performance on near- and far-OOD samples, we construct matched OOD subsets on CIFAR-100 [15] using its superclass hierarchy. Near-OOD classes belong to the same superclasses as the ID classes, whereas far-OOD classes belong to superclasses disjoint from those containing ID classes. As shown in Table V, our method achieves higher AUROC and lower FPR95 than CFSG-CLIP on both subsets. These results demonstrate improved discrimination between ID and challenging near-OOD samples.

Table V: Near- and far-OOD detection on CIFAR-100.

Method	Near-OOD AUROC (%) ↑	Far-OOD AUROC (%) ↑	AUROC gap (pp) ↓	Near-OOD FPR95 (%) ↓	Far-OOD FPR95 (%) ↓
CFSG-CLIP	64.98	82.38	17.40	89.65	76.70
Ours	**71.09**	**84.65**	**13.56**	**87.60**	**68.94**

C.7 Relationship to the CLIP Textual Embedding Space

We further analyze the relationship between enriched textual embeddings and the original CLIP textual embedding space. We measure the cosine similarity between each enriched textual embedding and its corresponding base textual embedding, and use Pearson and Spearman correlations to assess the preservation of inter-class relationships. As shown in Table VI, the Pearson and Spearman

Table VI: Relationship between enriched textual embeddings and the original CLIP text embedding space across four fine-grained datasets.

Dataset	Cosine sim.	Pearson	Spearman
Stanford Dogs	0.561	0.819	0.818
Stanford Cars	0.408	0.787	0.770
CUB-200-2011	0.284	0.801	0.764
FGVC-Aircraft	0.274	0.794	0.784

correlations suggest that the enriched textual embeddings largely preserve the inter-class semantic structure of the original CLIP text embeddings.

C.8 Analysis of Cross-Modal Injection Strategies

We further compare our method with a straightforward alternative cross-modal injection strategy. For each input image, this alternative directly concatenates its visual feature with the corresponding textual feature for classification, without using the alignment module, proposed memory bank, or fusion module. As shown in Table VII, single-image direct feature concatenation achieves 89.51% ID classification accuracy, while our design improves the result to 92.59%. This comparison indicates that the improvement does not come from simply appending visual features, but from the proposed aggregation and fusion design.

Table VII: Comparison with direct per-image feature concatenation on CUB-200-2011 under the 20-label setting.

Method	ID Acc. (%)	Bal. Acc. (%)
Direct per-image feature concatenation	89.51	88.65
Ours	**92.59**	**91.79**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are consistent with the method description in Section 3 and the experimental results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the main limitation of our method in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Our paper does not contain theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the proposed method in Section 3, and provide the training pipeline and implementation details in Section 4.1, which together support reproduction of the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is publicly available at `https://github.com/njustkmg/NeurIPS26-MDCEE`, and the experiments use public benchmark datasets.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All implementation details are described in Section 4.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All main results are reported as mean $\pm$ standard deviation over three runs with different random seeds, as stated in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides computer resource information in Appendix C.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have checked the NeurIPS code of ethics for our paper.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our paper focuses on a general fine-grained OSSL method, and does not specifically deal with broader societal impact considerations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper does not release high-risk generative models, scraped datasets, or other assets that would require special safeguards for responsible release.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The assets (datasets, code, models) used in the paper are open source, and our use follows the relevant protocols.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code we released for our paper contains related documents.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper doesn't involve the crowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper doesn't involve the potential risks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Not used at all.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.